


Article

A Method for Predicting Tool Remaining Useful Life: Utilizing BiLSTM Optimized by an Enhanced NGO Algorithm

Jianwei Wu ^{1,2}, Jiaqi Wang ² and Huanguo Chen ^{2,*} 

¹ School of Intelligent Manufacturing, Lishui Vocational & Technical College, Lishui 323000, China; 202210501016@mails.zstu.edu.cn

² School of Mechanical Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China; 202130605293@mails.zstu.edu.cn

* Correspondence: hgchen@zstu.edu.cn

Abstract: Predicting remaining useful life (RUL) is crucial for tool condition monitoring (TCM) systems. Inaccurate predictions can lead to premature tool replacements or excessive usage, resulting in resource wastage and potential equipment failures. This study introduces a novel tool RUL prediction method that integrates the enhanced northern goshawk optimization (MSANGO) algorithm with a bidirectional long short-term memory (BiLSTM) network. Initially, key statistical features are extracted from collected signal data using multivariate variational mode decomposition. This is followed by effective feature reduction, facilitated by the uniform information coefficient and Mann–Kendall trend tests. The RUL predictions are subsequently refined through a BiLSTM network, with the MSANGO algorithm optimizing the network parameters. Comparative evaluations with BiLSTM, BiGRU, and NGO-BiLSTM models, as well as tests on real-world datasets, demonstrate this method's superior accuracy and generalizability in RUL prediction, enhancing the efficacy of tool management systems.

Keywords: bidirectional long short-term memory; enhanced northern goshawk optimization; remaining useful life prediction; tool wear

MSC: 68T20; 68T07



Citation: Wu, J.; Wang, J.; Chen, H. A Method for Predicting Tool Remaining Useful Life: Utilizing BiLSTM Optimized by an Enhanced NGO Algorithm. *Mathematics* **2024**, *12*, 2404. <https://doi.org/10.3390/math12152404>

Academic Editors: Rui Peng, Kaiye Gao and Wen Zhang

Received: 2 June 2024

Revised: 29 July 2024

Accepted: 30 July 2024

Published: 2 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A tool is an integral part of machine tools and one of the components most susceptible to wear. It has a utility threshold, defined as the level of wear at which the quality of machined components is negatively impacted. Once this wear threshold is reached, a tool can no longer be used effectively. The time it takes for a tool to reach this failure threshold from its current cutting state is known as the remaining useful life (RUL). Understanding the tool wear threshold and RUL is crucial for maintaining production quality and efficiency [1]. Prematurely replacing tools before their failure leads to escalated costs and elongated manufacturing durations [2]. On the other hand, procrastinating their replacement can drastically undermine product quality and production efficiency [3,4]. Hence, the creation of a dependable and precise tool RUL prediction system is critically imperative.

Existing methods for predicting the RUL of cutting tools mainly fall into two categories: model-based and data-driven methods [5]. Model-based methods, grounded in a deep understanding of tool wear and degradation mechanisms, utilize physical or mathematical models to depict a tool's life degradation process [6]; however, these methods face limitations due to the influence of various factors creating uncertainty in parameter estimation and their inability to update measurement data in real time, which restrict their practical applicability [7].

As artificial intelligence advances, machine learning approaches that leverage signal features are gaining prominence in predicting the RUL of tools. Leveraging the formidable

feature learning prowess of machine learning models, a multitude of researchers have deployed algorithms like the BP neural network, support vector machine, and random forest for tool RUL prediction [8–10]. While these methods have demonstrated effectiveness, they frequently fail to adequately capture the temporal dependencies critical in extensive and complex industrial datasets. It is thus pivotal to delve into sophisticated algorithms that capitalize on the nuances of time series data, such as recurrent neural networks (RNNs) and their derivatives, which endow an essential edge in enhancing predictive accuracy. Illustratively, Yu et al. [11] crafted an attention–LSTM model predicated on cutting force signals for tool life prognostication. Rabah et al. [12] conceived a CNN–BiLSTM model that integrates signals of cutting force, vibration, and temperature for the same purpose. Nevertheless, the effective tuning of key parameters within a model still merits additional investigation.

Moreover, it is essential to highlight the importance of selecting a suitable optimization algorithm for determining a model’s ideal hyperparameters, a step that is vital for augmenting prediction accuracy [13,14]. The northern goshawk optimization (NGO) algorithm, noted for its straightforward structure and high precision in convergence, is extensively employed in optimizing parameters within machine learning models. For example, Yang et al. [15] employed the NGO algorithm to adaptively search for LSTM’s hyperparameters, yielding adept predictions of short-term runoff trends. Similarly, Zhong et al. [16] utilized the NGO algorithm to fine-tune the hyperparameters of the BiLSTM network effectively, which enhanced the accuracy of facial expression recognition. Nonetheless, the NGO algorithm suffers from suboptimal convergence precision and a propensity for local optima, attributed to its stochastic initialization and greedy paradigm for updating mechanisms, which can impede the optimization process.

Addressing the discussed challenges, this study introduces a predictive method that utilizes BiLSTM optimized by an enhanced NGO algorithm. Initially, the method employs multivariate variational mode decomposition (MVMD) on the acquired signal data to extract multiple intrinsic mode functions (IMFs), from which temporal and spectral features are extracted. Addressing the shortcomings of extant feature selection techniques and the distinct aspects of tool wear, a combined uniform information coefficient (UIC) and Mann–Kendall trend test (MKT) are applied to select features intimately linked with tool wear. The selected features are then fed into a BiLSTM network for feature learning, culminating in the RUL prediction. Concurrently, we incorporated two innovative enhancement strategies to address the previously identified limitations of the NGO algorithm. An enhanced northern goshawk optimization algorithm is utilized to calibrate the BiLSTM parameters optimally. The primary contributions of this study can be encapsulated in the following points:

An enhanced NGO algorithm has been introduced, incorporating a reverse learning strategy and a modified sine algorithm.

- (1) The convergence performance of the MSANGO algorithm has been evaluated through benchmark tests, with comparisons to the standard NGO algorithm and others.
- (2) The application of the MSANGO algorithm in selecting key parameters for the BiLSTM model significantly enhanced its predictive accuracy.

The remainder of this paper is structured as follows: Section 2 discusses data preprocessing, including signal feature extraction and selection. Section 3 elaborates on the theoretical underpinnings of the MSANGO–BiLSTM model. Section 4 elucidates the experimental design pertinent to our study and delves into a comprehensive analysis of the results. Section 5 encapsulates the concluding remarks.

2. Data Preprocessing

2.1. Extraction of Signal Features

In the machining process, signals typically display nonlinear and non-stationary traits. MVMD is a signal processing technique that decomposes complex multidimensional signals into several intrinsic mode functions (IMFs). Each IMF represents different frequency components of the original signal, helping to reveal the intrinsic characteristics and patterns [17].

This method is particularly suitable for processing non-stationary and nonlinear signals. In this study, a six-layer decomposition was conducted on signal data across seven channels to isolate various frequency components. Following signal decomposition, key features reflecting the state of the tool were extracted from both the time and frequency domains, revealing subtle changes in tool wear and providing a data foundation for subsequent monitoring models, thereby enhancing monitoring accuracy.

Time domain features offer direct information about changes in tool condition, as wear-induced alterations affect the fundamental statistical properties of the signals. Frequency domain features are derived from the power spectral density, uncovering shifts in energy distribution caused by tool wear or damage. These changes typically concentrate in specific frequency ranges, offering a solid physical basis for analysis. From each IMF, 17 time domain features and 5 frequency domain features were extracted, totaling 924 features (7 channels × 6 layers × (17 + 5)). Tables 1 and 2 provide detailed descriptions of these feature expressions and their physical meanings.

Table 1. Time domain feature expressions and their physical meanings.

Feature	Expression	Physical Meaning
Maximum value (Max)	$X_{\max} = \max\{x_i\}$	Maximum amplitude of the signal
Minimum value (Min)	$X_{\min} = \min\{x_i\}$	Minimum amplitude of the signal
Mean value (Mean)	$\bar{X} = \frac{1}{n} \sum_{i=1}^n x_i$	Central tendency of the signal
Peak-to-peak value (PP)	$X_p = X_{\max} - X_{\min}$	Range of the signal's amplitude
Absolute mean (Absm)	$X_{arv} = \frac{1}{n} \sum_{i=1}^n x_i $	Average energy of the signal
Variance (Var)	$X_{var} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$	Degree of dispersion of the signal
Standard deviation (Std)	$X_{\sigma} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$	Degree of fluctuation of the signal
Kurtosis (Kur)	$X_{kur} = \frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{X_{\sigma}} \right)^4$	Tendency of the signal to produce extreme values
Skewness (Ske)	$X_{ske} = \frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{X_{\sigma}} \right)^3$	Asymmetry of the signal's probability distribution
Root mean square (Rms)	$X_{rms} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}$	Overall energy or strength of the signal
Form factor (FF)	$X_{SF} = \frac{X_{rms}}{X_{arv}}$	Waveform shape characteristics of the signal
Crest factor (CF)	$X_{CF} = \frac{X_{\max}}{X_{rms}}$	Relative magnitude of the signal's peak value
Impulse factor (IF)	$X_{IF} = \frac{X_{\max}}{X_{arv}}$	Impact characteristics of the signal
Margin factor (MF)	$X_{mar} = \frac{X_{\max}}{X_r}$	Relative height of the signal's peak value
Root mean square amplitude (RMSA)	$X_{RMSA} = \frac{1}{n} \sum_{i=1}^n \sqrt{x_i^2}$	Distribution of the signal's amplitude
Kurtosis Factor (KF)	$X_{KF} = \frac{X_{kur}}{X_{var}^2}$	Peakedness of the signal's distribution
Skewness factor (SF)	$X_{SF} = \frac{X_{ske}}{X_{var}^2}$	Symmetry of the signal's distribution

Table 2. Frequency domain feature expressions and their physical meanings.

Feature	Expression	Physical Meaning
Frequency centroid (FC)	$FC = \left(\sum_{i=1}^n f_i p_i \right) / \left(\sum_{i=1}^n p_i \right)$	Central frequency of the signal’s spectrum
Mean square frequency (MSF)	$MSF = \left(\sum_{i=1}^n f_i^2 p_i \right) / \left(\sum_{i=1}^n p_i \right)$	Average energy level of the frequency components
Root mean square frequency (RMSF)	$RMSF = \sqrt{MSF}$	Overall energy level of the frequency distribution
Frequency variance (FV)	$VF = \left(\sum_{i=1}^n (f_i - F_c)^2 p_i \right) / \left(\sum_{i=1}^n p_i \right)$	Spread of the frequency distribution
Frequency standard deviation (FSD)	$RVF = \sqrt{VF}$	Variability of the frequency distribution

2.2. Feature Selection

2.2.1. Normalization

To eliminate the discrepancies in scale and numerical range between features, and to lay a solid foundation for the subsequent correlation and monotonicity analysis, this study employs the max-min normalization method to normalize the extracted features. Assuming that \bar{x} and x_i represent the normalized data and the original data, respectively, the formula for max-min normalization is as follows:

$$\bar{x} = \frac{x_i - \min\{x_i\}}{\max\{x_i\} - \min\{x_i\}}, \tag{1}$$

2.2.2. Uniform Information Coefficient (UIC)

During the tool wear process, the trend in signal characteristic changes is often not a simple linear relationship with the tool’s RUL curve. Introduced in 2022, the UIC is a correlation analysis method based on information theory [18]. The UIC is used to measure the correlation between two variables. Compared to the widely used Pearson correlation coefficient method and Spearman correlation coefficient method, the UIC can not only extract linear correlations between variables but also identify nonlinear correlations. Moreover, compared to the maximum information coefficient, the UIC effectively mitigates the influence of noise on correlation analysis, offering a lower computational cost, making it suitable for rapid dimensionality reduction in signal features with abundant data. The specific calculation method of the UIC is as follows:

For two sets of feature vectors, $X = [x_1, \dots, x_n]$ and $Y = [y_1, \dots, y_n]$, with a sequence length of n , the mutual information coefficient calculation model can be expressed as follows:

$$I_{MI}(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)}, \tag{2}$$

where $I_{MI}(X; Y)$ is the mutual information coefficient between X and Y ; $p(x, y)$ is the joint probability density function of X and Y ; and $p(x)$ and $p(y)$ are the marginal probability density functions of X and Y , respectively.

Using a uniform partitioning method, X and Y can be evenly divided into several segments, expressed as follows:

$$\begin{cases} l_x = \frac{x_{\max} - x_{\min}}{a}, 2 \leq a \leq 1 + \frac{n^{0.6}}{2} \\ l_y = \frac{y_{\max} - y_{\min}}{b}, 2 \leq b \leq 1 + \frac{n^{0.6}}{2} \end{cases} \tag{3}$$

where l_x and l_y represent the partition unit lengths of X and Y , respectively; x_{\max} and x_{\min} are the maximum and minimum values, respectively, of feature vector X ; y_{\max} and y_{\min} are the maximum and minimum values, respectively, of feature vector Y ; a and b denote

the number of segments for X and Y , respectively; and $n^{0.6}$ signifies the grid size of the partition, typically taken as the 0.6th power of the data volume [18].

2.2.3. Mann–Kendall Trend Test (MKT)

Cutting tools’ life degradation often shows time monotonicity, suggesting that features with strong monotonicity better capture a tool’s degradation path. The MKT is a non-parametric method that excels in analyzing time series with consistent monotonic trends. The MKT is used to identify trends in time series data without assuming a specific distribution, making it particularly useful for detecting monotonic trends. Given its wide applications in areas like hydrology and mechanical diagnostics [19,20], we use the MKT to identify key features with monotonic trends, enhancing the prediction of a tool’s RUL.

When analyzing with the MKT, the time series samples (x_1, x_2, \dots, x_n) consist of n independent, identically distributed random data. Suppose H_0 indicates that there is no trend in these sample data. The alternative hypothesis, H_1 , is a two-sided test, stating that for all $i, j \leq n$, and $i \neq j$, the distributions of x_i and x_j are different. The test statistic, S , is defined as follows:

$$S = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \text{sgn}(x_j - x_i), \tag{4}$$

$$\text{sgn}(x_j - x_i) = \begin{cases} 1 & x_j > x_i \\ 0 & x_j = x_i \\ -1 & x_j < x_i \end{cases} \tag{5}$$

where sgn is the sign function. When $n \geq 8$, S approximately follows a normal distribution with a mean of 0 and a variance defined as follows:

$$\text{Var}(S) = \frac{n(n-1)(2n+5) - \sum_{i=1}^g t_i(i-1)(2i+5)}{18}, \tag{6}$$

where g is the number of groups, with identical elements in the sequence grouped together; t_i is the range of the i -th node. By transforming S into Z_{mk} , Z_{mk} can approximate a standard normal distribution:

$$Z_{mk} = \begin{cases} \frac{S-1}{\sqrt{\text{Var}(S)}} & S > 0 \\ 0 & S = 0 \\ \frac{S+1}{\sqrt{\text{Var}(S)}} & S < 0 \end{cases} \tag{7}$$

Under the conditions of choosing a two-sided test and a significance level, α , if $|Z_{mk}| \geq Z_{1-\alpha/2}$, the null hypothesis, H_0 , is rejected. This implies that, at the significance level, α , the sample data exhibit a significant increasing or decreasing trend over time. For the two-sided test, when the absolute value of Z_{mk} is greater than or equal to 1.64, 1.96, or 2.57, it indicates that the test has passed at 90%, 95%, and 99% significance levels, respectively. The value of α is determined based on the sample size.

3. Enhanced-NGO-Algorithm-Optimized BiLSTM Model

3.1. Enhanced Northern Goshawk Optimization (MSANGO)

3.1.1. Northern Goshawk Optimization (NGO)

The NGO algorithm represents an innovative optimization algorithm, inspired by the predatory behavior exhibited by the northern goshawk. This method endeavors to pinpoint the optimal solution by perpetually refining the goshawk’s position [21]. The goshawk’s hunting tactics encompass two distinct stages: the prey-identification phase and the subsequent chase-and-evasion phase. The mathematical formulations underpinning the NGO algorithm, tailored to these specific hunting stages, are delineated below:

(1) Prey-Identification Phase

During the initial hunting stage of the northern goshawk, it selects a target at random and launches a rapid assault. The mathematical representation capturing the goshawk’s behavior in this phase is delineated below:

$$P_i = X_k, i = 1, 2, \dots, N; k = 1, 2, \dots, i - 1, \dots, N, \tag{8}$$

$$x_{i,j}^{new,p1} = \begin{cases} x_{i,j} + r(p_{i,j} - Ix_{i,j}), F_{P_i} < F_i \\ x_{i,j} + r(x_{i,j} - p_{i,j}), F_{P_i} \geq F_i \end{cases} \tag{9}$$

$$X_i = \begin{cases} X_i^{new,p1}, F_i^{new,p1} < F_i \\ X_i, F_i^{new,p1} \geq F_i \end{cases} \tag{10}$$

where P_i is the position of the prey targeted by the i -th northern goshawk; F_{P_i} is the objective function’s value, equating to the fitness score; k is a random integer within the $[1, N]$ range; and $X_i^{new,p1}$ portrays the updated state of the i -th goshawk, while $x_{i,j}^{new,p1}$ illustrates its renewed state in the j -th dimension, with $F_i^{new,p1}$ being the corresponding fitness metric. The variable r is a stochastic number confined to the $[0, 1]$ interval. Furthermore, I is a random variable that can take on the value 1 or 2. Both r and I serve as random determinants, facilitating the generation of sporadic NGO algorithm behaviors throughout the search and update processes.

(2) Chase-and-Evasion Phase

Upon the northern goshawk’s assault on its target, the prey instinctively seeks to flee. In this ensuing pursuit, the goshawk exhibits remarkable agility and speed, ensuring its capability to seize the prey in virtually any scenario. Given that this predatory act transpires within an attack radius denoted as R , the mathematical formulation for this secondary phase is articulated below:

$$x_{i,j}^{new,p2} = x_{i,j} + R(2r - 1)x_{i,j}, \tag{11}$$

$$R = 0.02 \left(1 - \frac{t}{T} \right), \tag{12}$$

$$X_i = \begin{cases} X_i^{new,p2}, F_i^{new,p2} < F_i \\ X_i, F_i^{new,p2} \geq F_i \end{cases} \tag{13}$$

where t is the present iteration count, while T is the maximum allowable iterations. $X_i^{new,p2}$ is the updated state of the i -th northern goshawk within the secondary hunting phase. Concurrently, $x_{i,j}^{new,p2}$ is the renewed state of the same goshawk in the j -th dimension during this phase. Lastly, $F_i^{new,p2}$ corresponds to the fitness metric associated with the new state.

3.1.2. Motivation for Improvement

In its iterative optimization journey, the NGO algorithm can exhibit challenges like suboptimal convergence precision and susceptibility to local optima. NGO’s approach to initializing the initial population relies on random initialization. While this tactic is simple, it risks compromising the diversity of the population’s solutions, thereby constraining the exploration scope. Furthermore, as deduced from Equation (10), NGO adopts a greedy paradigm for updating the population’s position during the prey-detection phase, which heightens the risk of the algorithm settling into local optima as iterations progress.

Given the foregoing analysis, an augmented NGO algorithm, named MSANGO, has been developed to enhance the original NGO’s effectiveness. Initially, a reverse learning strategy is employed for population initialization, enhancing diversity and aiding in detecting potential optimal solutions across an expanded solution space, thereby im-

proving global search capabilities. Further modifications include integrating a modified sine algorithm (MSA) within NGO’s prey-identification phase to refine search strategies, where the MSA incorporates adaptive weighting coefficients during position updates. This adjustment fosters a robust equilibrium between global exploration and localized exploitation. The MSANGO algorithm’s procedural framework is depicted in Figure 1, with the mathematical rationale for the enhanced strategies detailed subsequently.

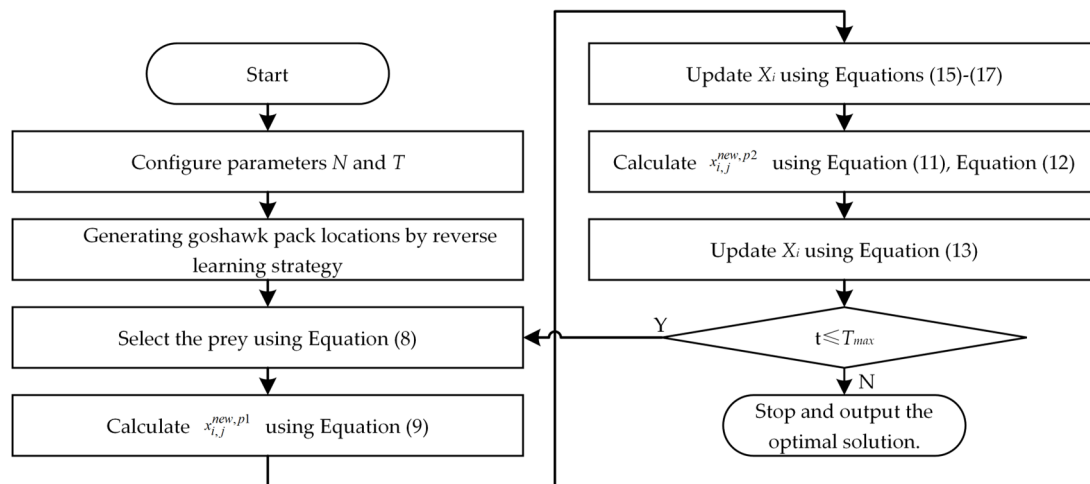


Figure 1. Flowchart of the MSANGO algorithm.

3.1.3. Reverse Learning Strategy

The reverse learning strategy is a heuristic for population initialization, enhancing diversity and exploration. It uses existing solutions to create ‘reverse’ counterparts, broadening search domain coverage. Recently, many researchers have adopted this method, achieving improved performance and faster convergence in various tasks [22,23].

Let the solution for individuals, i , of the northern goshawk be $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$, where $i = 1, 2, \dots, N$, and D is the dimension of the search. The reverse solution for an individual’s position is defined as follows:

$$x'_{i,j} = \delta \cdot (lb_j + ub_j) - x_{i,j}, \tag{14}$$

where δ is a stochastic variable uniformly distributed between 0 and 1. The term $x_{i,j}$ is the component of an elite individual’s, i ’s, solution in the j -th dimension. $x'_{i,j}$ is the reverse solution corresponding to $x_{i,j}$. The index, j , varies from 1 to D , which represents the dimensions of the solution space. The lower and upper bounds within the j th dimension are indicated by lb_j and ub_j , respectively. The population initialization process based on reverse learning is as follows:

Step1: Generate an initial population of northern goshawks, RP , with N individuals through random initialization.

Step 2: Apply reverse learning to create reverse solutions for each RP member, forming the reverse population, OP .

Step 3: Combine the RP and OP to form the new initial population, NP , with a total of $2N$ individuals.

3.1.4. Modified Sine Algorithm

The MSA leverages the sine function from mathematics for iterative optimization, demonstrating robust global search capabilities [24]. It incorporates an adaptive variable inertia weight coefficient during the position-updating process, enhancing local area searches

and facilitating an effective balance between global exploration and local exploitation. The position-updating formula of the MSA is presented below:

$$x_i(t + 1) = \omega_t x_i(t) + r_1 \times \sin r_2 \times [r_3 p_i(t) - x_i(t)], \tag{15}$$

where t is the current iteration number, and $x_i(t)$ is the i -th positional component of an individual, X , at the t -th iteration. Additionally, $p_i(t)$ refers to the i -th component of the optimal position variable during the same iteration. The function r_1 acts as a nonlinear decreasing function, while r_2 and r_3 are random numbers within the intervals $[0, 2\pi]$ and $[-2, 2]$, respectively.

The value of r_1 is configured using a nonlinear decreasing pattern, with changes in r_1 determined by a cosine function spanning the interval from 0 to π :

$$r_1 = \frac{\omega_{\max} - \omega_{\min}}{2} \cos \frac{\pi t}{T_{\max}} + \frac{\omega_{\max} + \omega_{\min}}{2}, \tag{16}$$

where ω_{\max} and ω_{\min} denote the upper and lower bounds of ω_t , respectively. The variable t indicates the current iteration, while T_{\max} specifies the total allowable iterations.

ω_t , an adaptive variable inertia weight, diminishes linearly with an increase in the iteration count:

$$\omega_t = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times \frac{t}{T_{\max}}, \tag{17}$$

3.2. MSANGO Algorithm Performance Evaluation

3.2.1. Settings for Algorithm Parameters

The MSANGO algorithm has undergone a comparative analysis against the sine cosine algorithm (SCA), whale optimization algorithm (WOA), and NGO algorithm. It is important to note that the parameter settings for these algorithms are based on the recommendations provided in the relevant scholarly literature. Specifically, in the SCA, $r_1 = 2 - 2t/t_{\max}$. For the WOA, the convergence factor linearly decreases from 2 to 0. In pursuit of robust result validation, each algorithm underwent 1000 iterations and 30 separate trials. The mean and standard deviation values from these 30 trials serve as the benchmarks for assessing convergence performance.

Experiments were performed on a system powered by an AMD R7-5800H CPU at 3.2 GHz, supported by 32 GB of RAM, and using the Windows 11 operating system. The simulations were executed using MATLAB version 2022b.

3.2.2. Benchmark Test Functions and Results

The MSANGO algorithm's efficacy is validated through simulations with six benchmark test functions. Unimodal functions F1 and F2 assess the algorithm's convergence speed and accuracy. Multimodal functions F3 and F4 evaluate its capability to circumvent local optima and achieve global optima. Fixed-dimension multimodal functions F5 and F6 investigate the algorithm's performance in addressing specific dimensional challenges. Table 3 details these functions, with ' n ' representing the dimensionality of the search space.

The convergence performance of the algorithm is evaluated through the mean and standard deviation of the fitness values. A lower mean indicates greater precision in algorithmic convergence, while a reduced standard deviation signifies enhanced stability. The calculations for mean and standard deviation are defined as follows:

$$\text{Mean} = \frac{1}{M} \sum_{i=1}^M \text{fitness}(i), \tag{18}$$

$$\text{Std} = \sqrt{\frac{1}{M} \sum_{i=1}^M (\text{fitness}(i) - \text{Mean})^2}, \tag{19}$$

where fitness(*i*) denotes the fitness value of the *i*-th test, while *M* refers to the total number of experimental runs.

Table 3. Benchmark test functions.

Expression	Dimension	Search Space	Optimal Value
$F_1(x) = \sum_{i=1}^n x_i^2$	30	[−100, 100]	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[−10, 10]	0
$F_3(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[−5.12, 5.12]	0
$F_4(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	[−32, 32]	0
$F_5(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right) \cos x_1 + 10$	2	[−5, 5]	0.398
$F_6(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2\right)$	3	[1, 3]	−3.86

Table 4 illustrates the convergence curves of various algorithms on benchmark test functions. The MSANGO algorithm demonstrates superior performance on unimodal functions, attaining the lowest average values and enhanced stability. In the realm of multimodal functions, although all algorithms exhibit commendable performance, MSANGO stands out in terms of function F4, with exceptional adaptability and precision. For fixed-dimension multimodal functions, the algorithms’ performances are closely matched, which is particularly evident for function F5, where the results are almost indistinguishable. Notably, for function F6, the NGO algorithm maintains the greatest stability, whereas MSANGO exhibits slightly reduced performance.

Table 4. Comparative analysis of optimization outcomes for benchmark functions.

Function	Statistic	Algorithm			
		SCA	WOA	NGO	MSANGO
F1	Mean	0.0545	4.99×10^{-152}	2.03×10^{-178}	0
	Std	0.1696	2.49×10^{-151}	0	0
F2	Mean	1.98×10^{-5}	1.83×10^{-105}	2.14×10^{-92}	0
	Std	3.17×10^{-5}	4.75×10^{-105}	7.73×10^{-105}	0
F3	Mean	18.1164	0	0	0
	Std	22.1622	0	0	0
F4	Mean	15.1692	3.29×10^{-15}	5.77×10^{-15}	4.44×10^{-16}
	Std	7.9876	2.13×10^{-15}	1.78×10^{-15}	0
F5	Mean	0.3990	0.3979	0.3979	0.3979
	Std	0.0014	4.26×10^{-7}	0	1.42×10^{-9}
F6	Mean	−3.8547	−3.8597	−3.8628	−3.8594
	Std	0.0024	0.0036	2.66×10^{-15}	0.0028

3.2.3. Analysis of the Convergence Process

Figure 2 illustrates the optimization performance of the MSANGO algorithm alongside other algorithms for various test functions, providing a visual comparison. For unimodal functions, the SCA, WOA, and NGO algorithm demonstrate lower optimization accuracy and slower convergence. In contrast, the MSANGO algorithm achieves rapid convergence to the theoretical optimum within 500 generations for function F1. With multimodal

functions, the MSANGO algorithm reaches the theoretical optimum within just 30 generations for function F3. Although the MSANGO algorithm does not fully converge to the theoretical optimum for function F4, it exhibits superior optimization precision and lower variability compared to the other three algorithms. For fixed-dimension multimodal functions, while all algorithms achieve convergence to the theoretical optimum for function F5, the MSANGO algorithm does so more swiftly. The performance for function F6 shows the MSANGO algorithm closely aligning with the other algorithms.

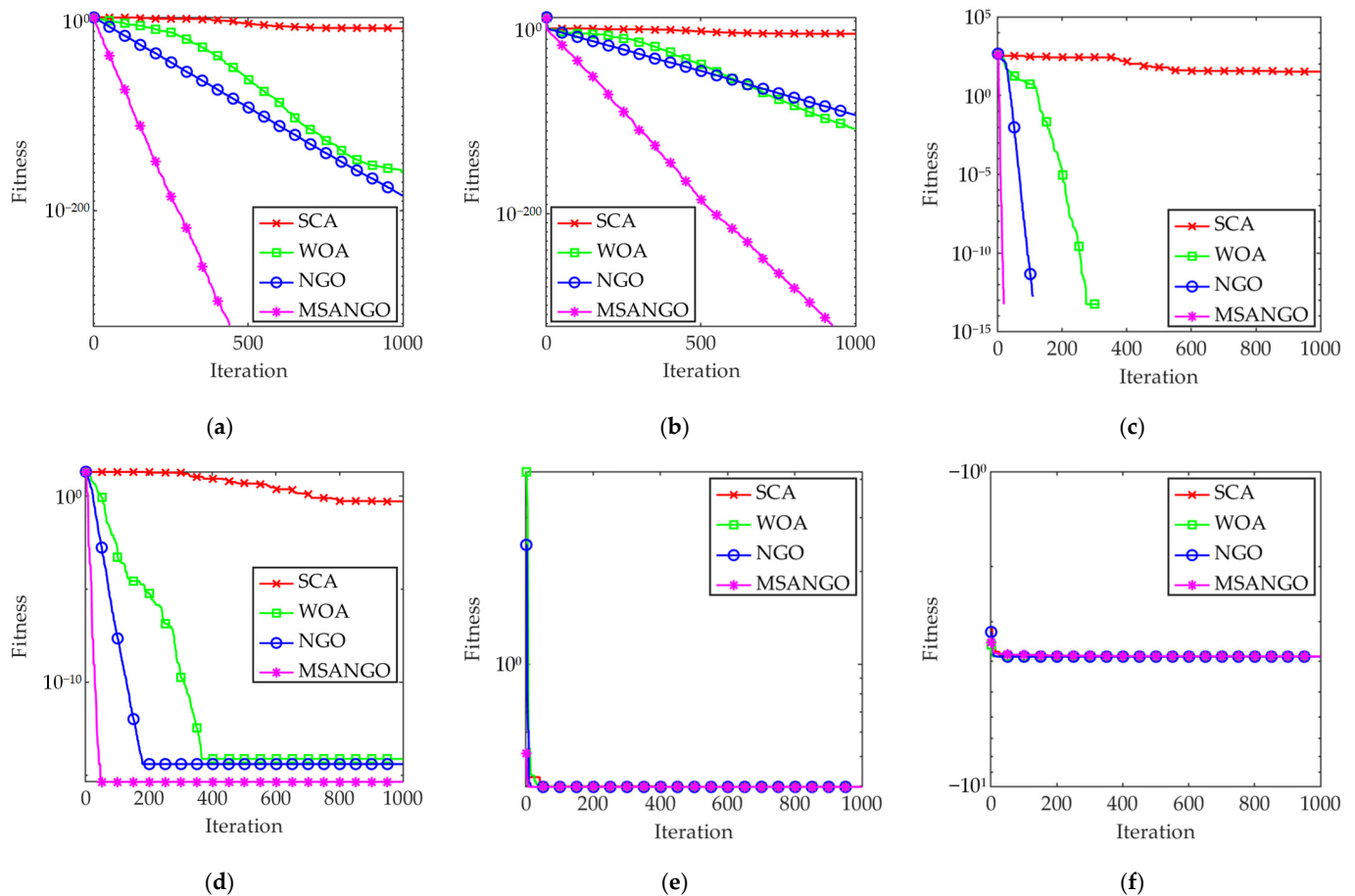


Figure 2. Flowchart of the MSANGO algorithm. (a) F1 (b) F2 (c) F3 (d) F4 (e) F5 (f) F6.

Consequently, the MSANGO algorithm consistently demonstrates outstanding performance across both unimodal and multimodal functions, characterized by its rapid convergence and capacity for in-depth exploitation. This highlights the algorithm’s effective balance between global exploration and local exploitation capabilities.

3.2.4. Analysis of Time Complexity

Time complexity serves as a crucial metric for evaluating the convergence speed of optimization algorithms. This study employs Big O notation for the computation of time complexity [25]. The Big O notation is derived via the following steps: initially, replace all additional constants in runtime with the constant 1; subsequently, retain only the highest-order term in the modified runtime expression; and finally, if a non-unit highest-order term exists, its multiplicative constant is removed, resulting in a Big O expression. Suppose that t denotes the time required to evaluate the objective function. The time complexity analysis for the NGO algorithm starts with the initialization phase, where generating initial population positions incurs $O(N \times dim)$. Each iteration involves exploration and exploitation phases, each having a time complexity of $O(N \times dim + N \times t)$. Consequently, the total time complexity for the NGO algorithm is $O(N \times dim + T \times (N \times dim + N \times t))$.

As the MSANGO algorithm introduces no new loops and retains the original sequence in its computations, its time complexity also adheres to $O(N \times dim + T \times (N \times dim + N \times t))$.

The algorithms were executed on all functions for 30 trials, with each trial consisting of 1000 iterations, to gather running times, which were then averaged across the trials, as detailed in Table 5. For every function, from F1 to F6, the MSANGO algorithm consistently demonstrated shorter running times compared to the NGO algorithm, underscoring its superior efficiency on all tested functions.

Table 5. Comparative analysis of CPU execution times for the NGO and MSANGO algorithms.

Algorithm	F1	F2	F3	F4	F5	F6
NGO	0.0550	0.0602	0.0656	0.0666	0.0343	0.0579
MSANGO	0.0342	0.0399	0.0412	0.0439	0.0178	0.0418

3.3. MSANGO-BiLSTM Model

LSTM, an especially proficient variant of RNNs, has proven adept at mitigating the gradient explosion or vanishing challenges inherent to basic RNNs [26]. Within the LSTM architecture, three distinct gates are incorporated: input, forget, and output. The input gate assesses the retention level of the current candidate state’s information; the forget gate modulates the amount of information discarded from the prior internal state; and the output gate governs the volume of information relayed from the internal to the external state. As depicted in Figure 3, a standard LSTM cell is presented, with it symbolizing the input gate; g_t is the current candidate state; f_t is the forget gate; O_t is the output gate; and X_t is the present-time input. Here, σ is the sigmoid function, bounded between (0, 1), and \tanh is the hyperbolic tangent, a prevalent activation function.

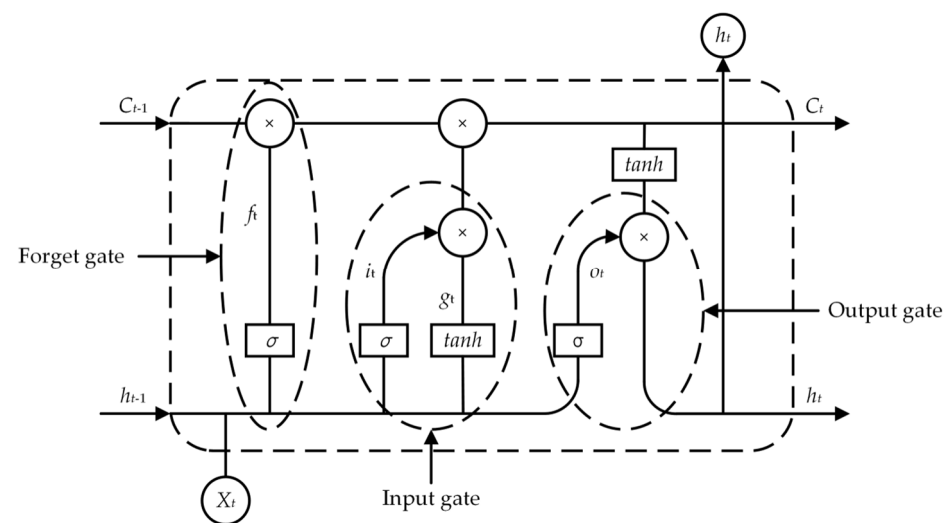


Figure 3. Cell structure diagram of LSTM.

BiLSTM incorporates two distinct LSTM hidden layers, enabling it to process sequence data bidirectionally—capturing both historical and forthcoming information. This bidirectional paradigm, offering an edge over the traditional unidirectional LSTM, inherently possesses enhanced predictive potential. As depicted in Figure 4, a detailed schematic of BiLSTM is presented: x_t is the BiLSTM input, \vec{h}_t is the forward propagation hidden layer’s output, \overleftarrow{h}_t corresponds to the backward propagation hidden layer’s output, and y_t represents the BiLSTM output.

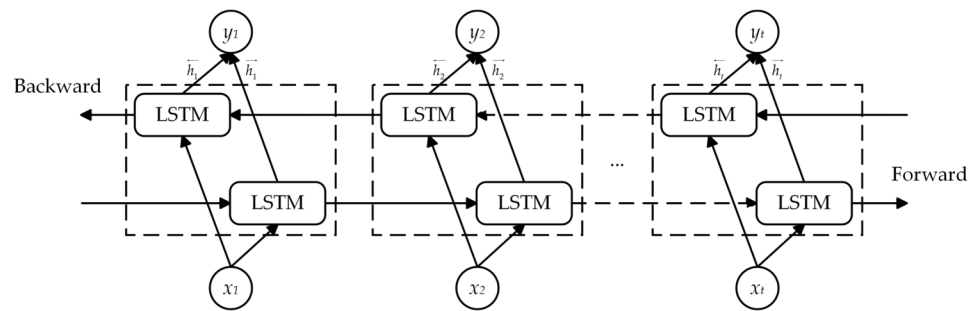


Figure 4. Schematic diagram of BiLSTM.

In the BiLSTM framework, pivotal parameters such as the initial learning rate, neuron count in the hidden layer, and regularization coefficient play a substantial role in dictating the efficacy of time series predictions; however, the field lacks a comprehensive theoretical foundation with which to guide the calibration of these parameters, often relegating their determination to heuristic methods [27,28]. Considering these factors, this study utilizes the MSANGO algorithm to optimize three specific parameters. The average root mean square error from two tools within the training dataset serves as the fitness function for the MSANGO algorithm, guiding the identification of optimal parameter settings.

4. Application of Tool RUL Prediction

4.1. Description of the Experiment

For the empirical validation of the methodology delineated in this manuscript, the PHM 2010 dataset, originating from the 2010 high-speed CNC machine tool health prediction competition organized by the Prognostics and Health Management Society in New York, was selected [29]. This dataset comprehensively documents the lifecycle of six cutting tools from their inception to their wear-out stage, providing real-time data from multiple sensors. It is highly relevant to tool condition monitoring (TCM) scenarios, as it records tool wear status, facilitating the development and validation of tool wear prediction models [30,31]. The data emanate from experiments conducted on the Rödgers Tech RFM760 CNC machine tool. The experimental parameters were set as follows: a spindle speed of 10,400 revolutions per minute, a feed rate of 1555 mm per minute, a Y-axis cutting depth of 0.125 mm, and a Z-axis cutting depth of 0.2 mm. Within this experimental framework, the study utilized a ball nose carbide milling cutter for face milling operations on a workpiece with a hardness rating of HRC52.

In the experimental phase, the study incorporated a Kistler 9265B triaxial dynamometer positioned between the workpiece and the machining platform for the quantification of cutting force signals. Concurrently, vibrations across three axes were monitored using three Kistler 8636C accelerometers. Additionally, a Kistler 8152 acoustic emission sensor was affixed to the workpiece to detect acoustic signals. The aggregation of these signals was facilitated by an NI PCI1200 data acquisition card, encompassing seven signal channels and operating at a sampling frequency of 50 kHz. A schematic representation of this experimental setup is depicted in Figure 5.

This work focused on three specific milling cutters (C1, C4, and C6) from the dataset. Each of these milling cutters underwent 315 identical cutting cycles under standardized machining conditions. After each cutting session, the flank wear of the tools was measured offline using a LEICA MZ12 microscope. For the purposes of this study, two of these tools were allocated for training the model, while the third served as the test subject, thereby facilitating an assessment of the model's regression and generalization capabilities. The configuration of the experimental groups is detailed in Table 6. Furthermore, to filter out the non-representative data typically generated during the initiation and cessation phases of the milling process, a subset of one thousand data points, centered around the midst of each cutting cycle, was earmarked for detailed analysis.

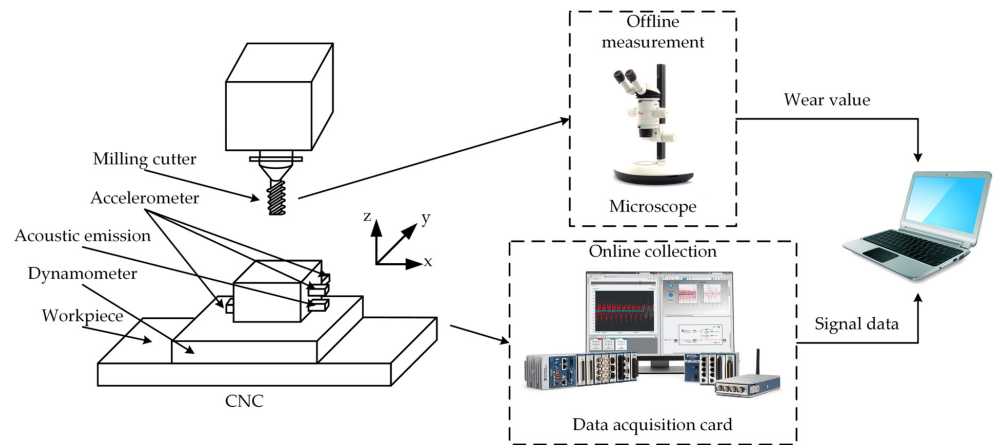


Figure 5. Schematic diagram of the experimental setup.

Table 6. Configuration of the experimental groups.

Experimental Group	Training Set	Testing Set
D1	C4, C6	C1
D2	C1, C6	C4
D3	C1, C4	C6

4.2. Results of Feature Selection

Only the top 10% of features, ranked by their UIC values, were retained to reduce model complexity, enhance computational efficiency, and ensure the inclusion of the most informative features, thereby improving the model’s generalization and stability. The final selection step involved employing the MKT to isolate features exhibiting pronounced monotonic trends. For this study, given a sample size exceeding 100, the significance level, α , was established at 0.05. This means that a feature is considered statistically significant at a 95% confidence level if the $|Z_{MK}|$ is 1.96 or higher.

For a lucid illustration of the robust correlation between the optimized feature set and the tool’s RUL, Figure 6 depicts the frequency at which various features appear across the experimental groups, alongside the average UIC value and the mean $|Z_{MK}|$ value associated with each feature.

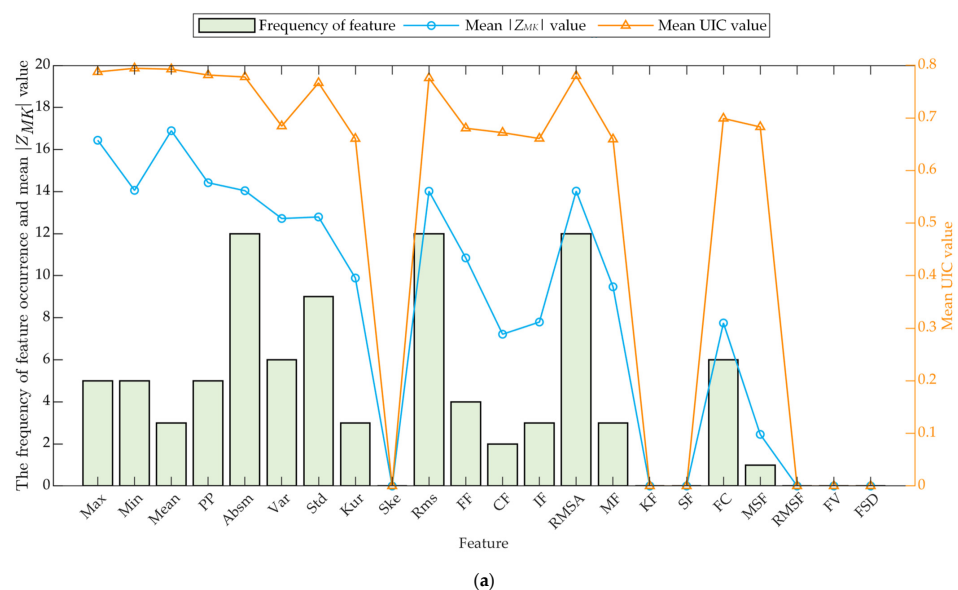


Figure 6. Cont.

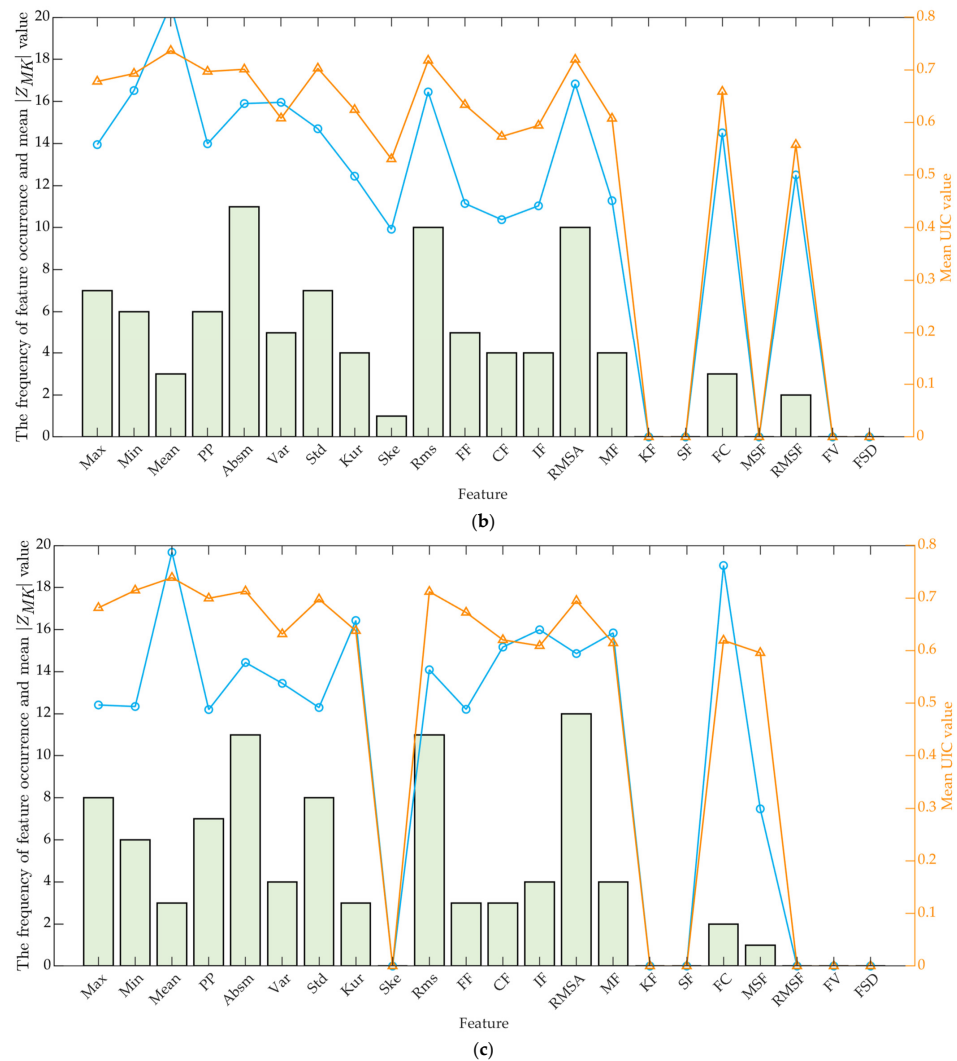


Figure 6. The frequency at which various features appear across the experimental groups, alongside the average UIC values and the mean $|Z_{MK}|$ values associated with each feature: (a) D1; (b) D2; and (c) D3.

It is observable that, in all three experimental groups, the frequency of features such as Absm, Rms, and RMSA significantly exceeds that of others, indicating their greater importance and stability in predicting the tool’s RUL. In contrast, the features KF, SF, FV, and FSD have a frequency of occurrence of zero, while Ske and RMSF do not appear in two experimental groups, suggesting their lesser efficacy in representing the tool life degradation process. Furthermore, it is observed that for the majority of features, a strong monotonicity often coincides with high correlation, and vice versa.

4.3. Configuration of BiLSTM Network Structure and Parameters

We developed a deep learning framework incorporating BiLSTM layers, dropout layers, a fully connected layer, and a regression layer, as depicted in Figure 7. The input layer receives sequence data reflecting the number of signal channels, enabling the handling of variable-length time series. Following the input layer, two BiLSTM layers extract long-term dependencies and intricate patterns from the data. To mitigate overfitting, each BiLSTM layer is followed by a dropout layer with a dropout rate of 0.3, enhancing the model’s generalizability. The fully connected layer projects the double-layer BiLSTM’s feature representation into a one-dimensional output space. The regression output layer, employing a sigmoid activation function for predictions, utilizes root mean squared error as its loss metric to prioritize minimizing discrepancies between predicted and actual values.

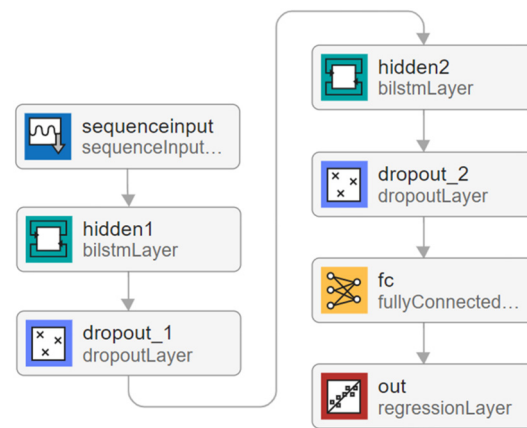


Figure 7. Schematic of the BiLSTM network structure.

Training parameters include the use of the Adam optimizer, a maximum of 300 epochs, and a batch size of eight. A gradient threshold of one prevents gradient explosion. The initial learning rate, the number of neurons in the hidden layers, and the L2 regularization coefficient are optimized using the MSANGO algorithm proposed in this paper. The learning rate, starting from an initial value determined by the MSANGO algorithm, is adjusted by a piecewise constant strategy, decreasing by 20% every 10 epochs to further enhance training efficiency.

4.4. Assessment of Prediction Performance

In this work, we employ a trio of evaluation criteria, mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE), to holistically gauge the predictive prowess of the model across diverse facets. The RMSE serves as an indicator of the model's adeptness at mitigating substantial errors, shedding light on its inherent robustness. Conversely, both MAE and MAPE offer a lucid depiction of the precision of the predictions. A diminutive value across these metrics signifies enhanced model efficacy. The computational expressions for these criteria are delineated below:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (20)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (21)$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\%, \quad (22)$$

where y_i is the true value, \hat{y}_i is the forecasted value, and n is the total sample count.

Figure 8 illustrates the convergence curves for the NGO and MSANGO algorithms on the training dataset. Observations from Figure 8 indicate that the MSANGO algorithm consistently outperforms the NGO algorithm in terms of convergence across all experimental groups. Notably, in the D2 and D3 groups, the initial fitness values of the MSANGO algorithm are substantially superior to those of the NGO algorithm. This enhanced performance is largely attributable to the MSANGO algorithm's refined approach, which includes initializing populations through a reverse learning strategy. Furthermore, the incorporation of the modified sine algorithm significantly enhances the MSANGO algorithm's search efficacy. This improvement enables the algorithm to adeptly avoid local optima, thereby facilitating more effective navigation of the search space and accelerating convergence to optimal solutions.

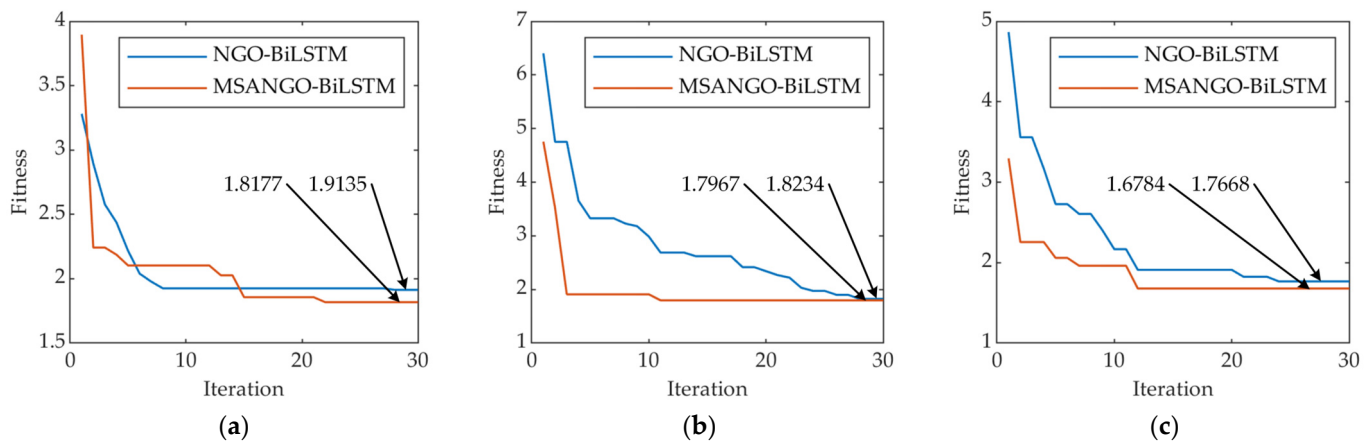


Figure 8. Convergence trajectories of the NGO and MSANGO algorithms. (a) D1 (b) D2 (c) D3.

4.5. Comparative Analysis with Leading-Edge Prediction Model

This study compares the predictive performance of the proposed MSANGO-BiLSTM regression prediction model with other models employed in previous research, including BiLSTM [32] and BiGRU [33]. Both the BiLSTM and BiGRU models are variants of recurrent neural networks, whereas the NGO-BiLSTM model serves to illustrate the benefits of parameter optimization via an optimization algorithm and underscores the necessity for enhancements to the NGO algorithm.

The configurations for the BiLSTM and BiGRU models are elaborately outlined in Table 7. For the NGO-BiLSTM and MSANGO-BiLSTM models, the optimization algorithm’s population size is established at 10, with a maximum iteration limit of 30. The range for optimizing the initial learning rate is set between 0.001 and 0.1, the neuron count within the hidden layers ranges from 50 to 200, and the regularization coefficient spans from 0.001 to 0.1. Detailed descriptions of the BiLSTM network structure and parameters are provided in Section 4.3.

Table 7. Parameter settings of comparison models.

Model	Hidden Layers	Epoch	Batch Size	Neurons	Initial Learning Rate	Optimizer
BiLSTM	1	2750	16	110	0.01	Adam
BiGRU	1	100	16	3	0.001	Adam

Figures 9–11 compare the predictive performance of various models against actual values across three experimental groups. The BiGRU model demonstrates suboptimal fitting performance, attributable to its limited number of neurons and reduced iteration count. The BiLSTM and NGO-BiLSTM models, performing comparably, effectively track the RUL’s decline but sometimes tend to pessimistically estimate a tool’s RUL towards the end, predicting fewer cuts than is actually the case. In contrast, the MSANGO-BiLSTM model excels across all three groups, demonstrating consistent stability and accuracy, particularly in the crucial late stage with minimal deviation from actual values. This underscores the MSANGO-BiLSTM model’s superior fitting ability in tool RUL prediction.

Table 8 and Figure 12 display the performance metrics for each model within the three experimental groups. The results demonstrate that, in the context of predicting tool RUL, the three BiLSTM models generally exhibit superior predictive performance compared to the BiGRU model. Additionally, models that have been subjected to specialized optimization processes, such as MSANGO-BiLSTM and NGO-BiLSTM, achieve superior predictive accuracy compared to the standard BiGRU and BiLSTM frameworks. This highlights the significant role of custom-tailored model optimization in augmenting the effectiveness of predictions in specific applications.

Table 8. Evaluation metrics for the prediction results of each model.

Model	D1			D2			D3		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
BiLSTM	20.1574	26.7955	0.4220	10.4098	14.1462	0.1509	26.6519	32.7777	0.8976
BiGRU	52.2303	65.2801	0.9556	15.2669	19.2226	0.4036	25.9049	33.9597	0.4319
NGO-BiLSTM	22.5293	27.7053	0.6240	10.6165	12.5502	0.1747	23.4395	29.3256	0.3774
MSANGO-BiLSTM	18.6473	23.4638	0.2696	7.9252	10.4044	0.2854	18.5267	21.4623	0.3620

The BiGRU model demonstrates underwhelming performance across the three experimental groups. In comparison, the BiLSTM model significantly improves in terms of MAE, RMSE, and MAPE, showing average reductions of 30.11%, 29.61%, and 3.54%, respectively. The NGO-BiLSTM model matches the performance of the BiLSTM model in groups D1 and D2; however, it achieves reductions in MAE, RMSE, and MAPE of 12.05%, 10.53%, and 57.95%, respectively, in group D3.

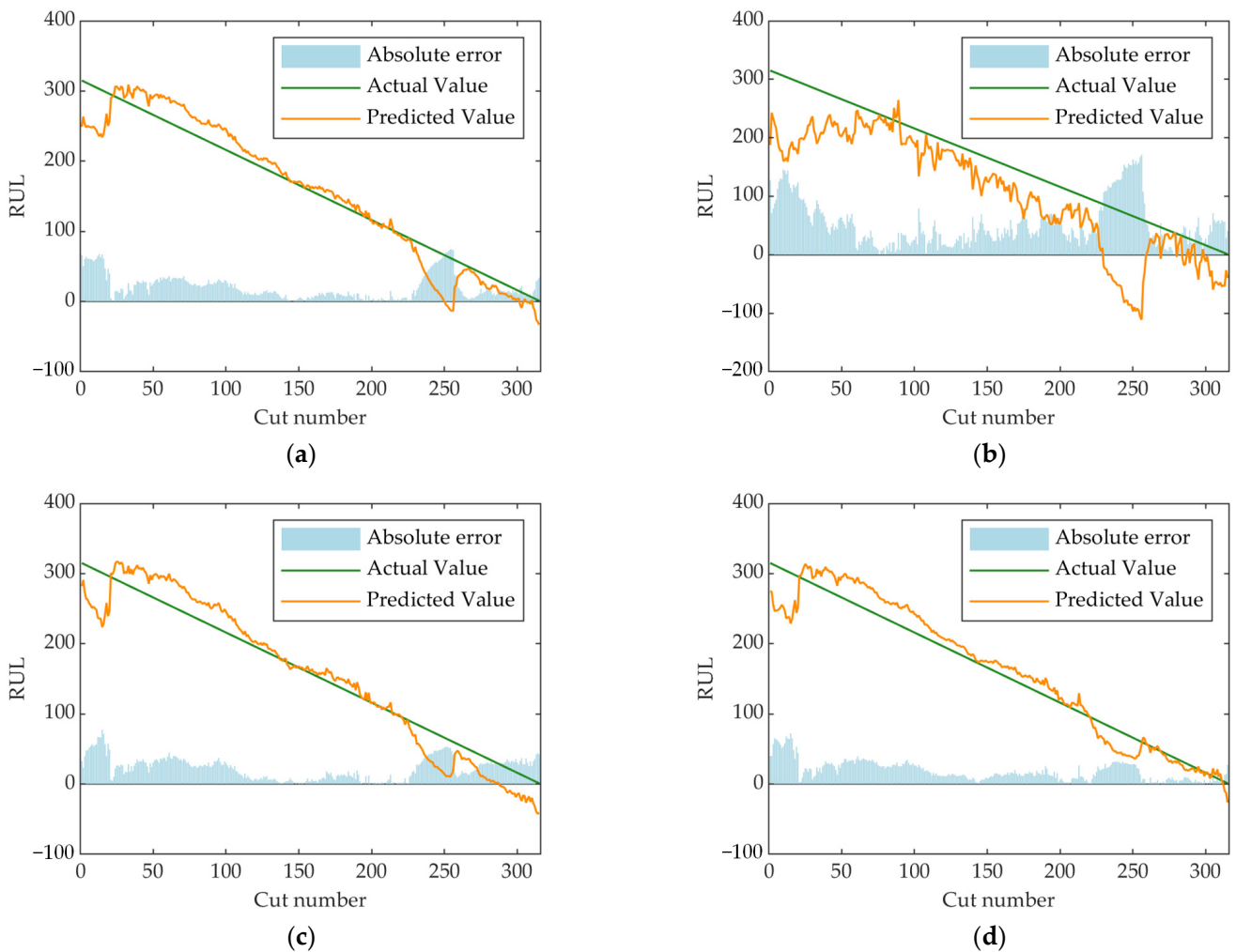


Figure 9. The predictive results of the four models on experimental group D1. (a) BiLSTM (b) BiGRU (c) NGO-BiLSTM (d) MSANGO-BiLSTM.

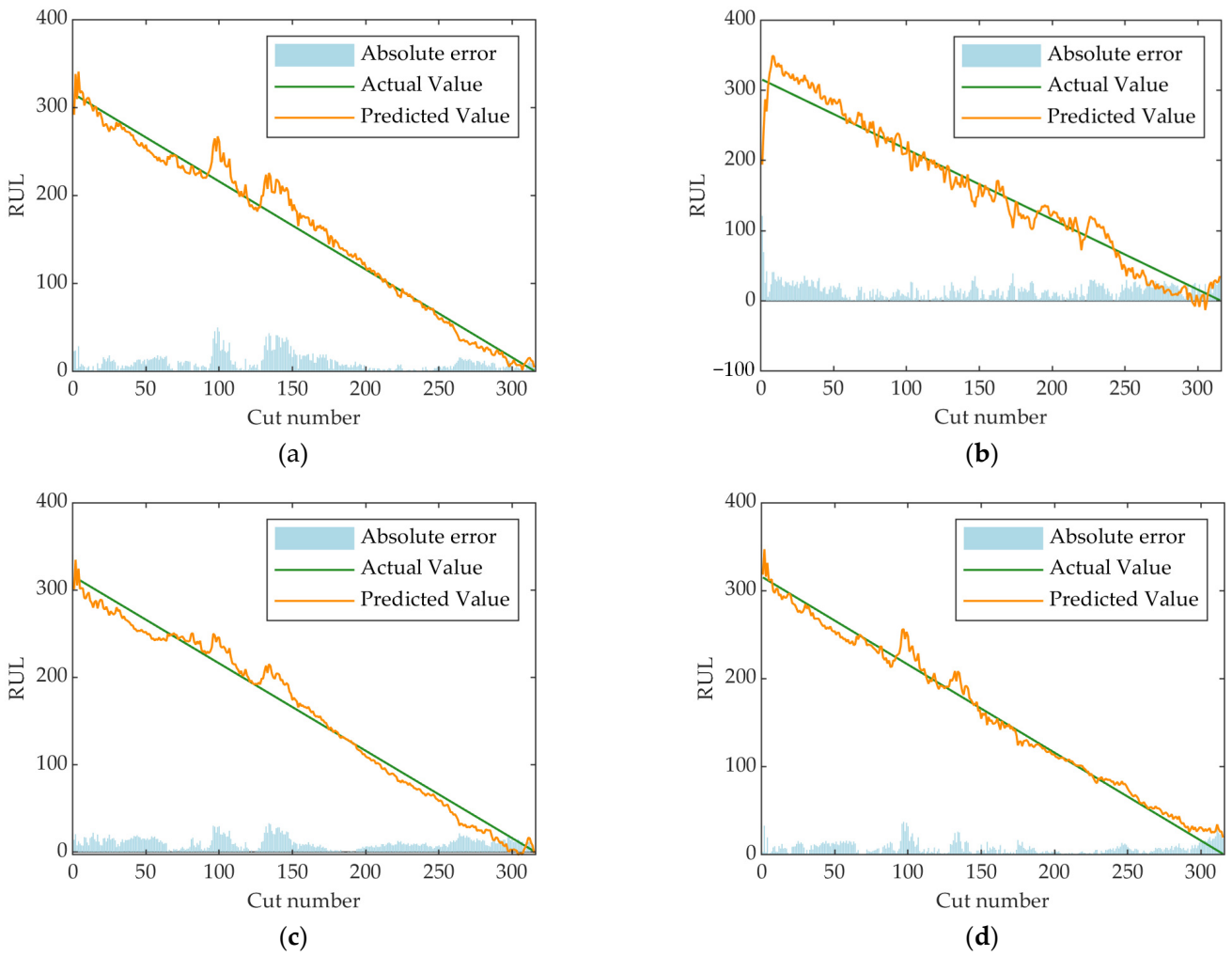


Figure 10. The predictive results of the four models on experimental group D2. (a) BiLSTM (b) BiGRU (c) NGO-BiLSTM (d) MSANGO-BiLSTM.

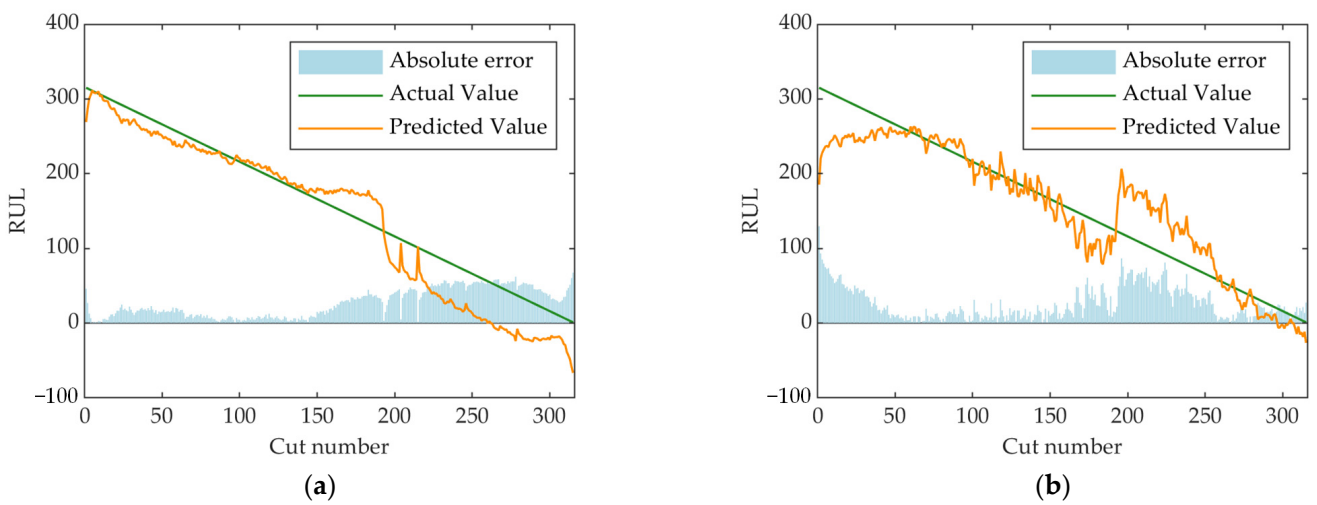


Figure 11. Cont.

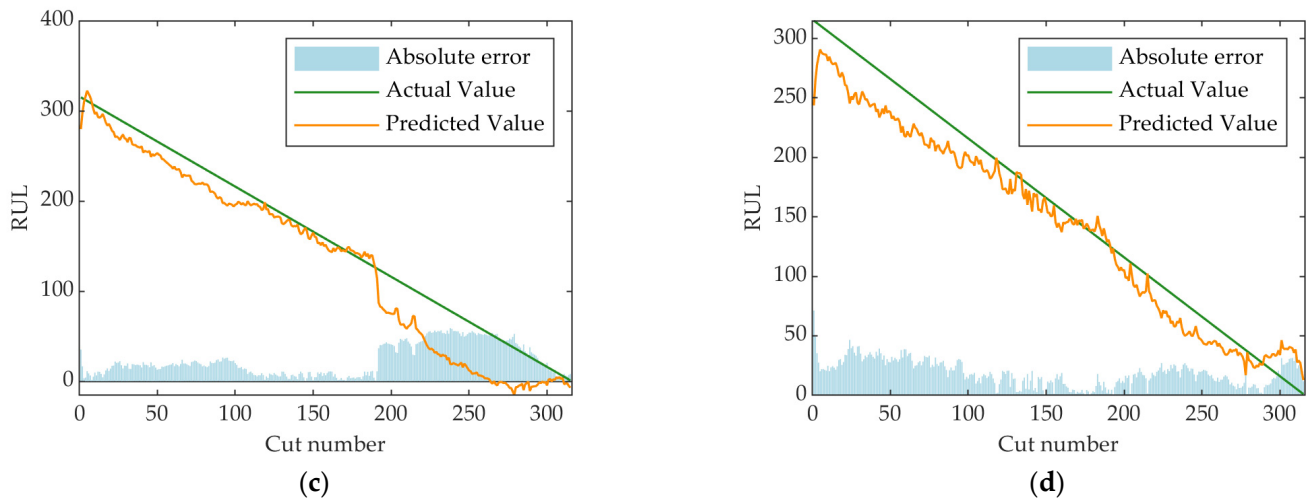


Figure 11. The predictive results of the four models on experimental group D3. (a) BiLSTM (b) BiGRU (c) NGO-BiLSTM (d) MSANGO-BiLSTM.

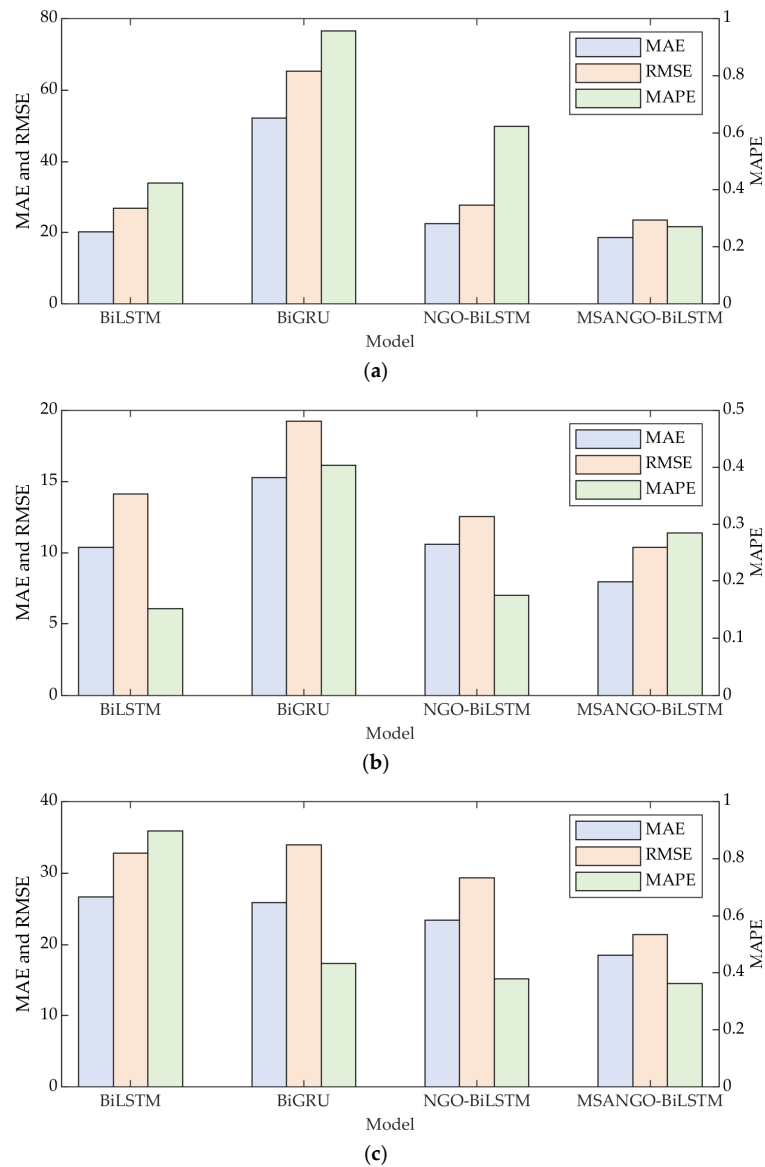


Figure 12. Comparative of evaluation metrics among each model. (a) D1 (b) D2 (c) D3.

Notably, the MSANGO-BiLSTM model excels in the MAE and RMSE metrics, which penalize larger errors more severely. Relative to the BiLSTM, BiGRU, and NGO-BiLSTM models, reductions in MAE and RMSE for the MSANGO-BiLSTM are at least 7.49% and 12.43%, indicating consistent stability and precision in its predictions. Although its MAPE is slightly elevated in group D2, it reduces significantly by at least 36.11% and 4.08% in groups D1 and D3, respectively. Overall, the MSANGO-BiLSTM model's superior performance underscores its strong generalization capabilities and stability, affirming its efficacy in precise data approximation.

Table 9 demonstrates that models optimized by optimization algorithms exhibit a marked increase in training time, entailing extra computational resources and time expenditures. Future strategies could leverage advanced computational hardware and implement parallel as well as distributed training methods to enhance training efficiency substantially. Moreover, the MSANGO algorithm introduced in this study decreases training costs by at least eight percent relative to the NGO algorithm, providing a more efficient alternative. Despite the increase in training duration, the considerable gains in model performance justify the additional investment in training time.

Table 9. Training time of each model.

Model	Training Time/s		
	D1	D2	D3
BiLSTM	223.6069	224.2265	219.1590
BiGRU	17.8720	13.7725	14.4773
NGO-BiLSTM	4396.4600	4295.8060	4404.4936
MSANGO-BiLSTM	4036.3645	3952.0609	4028.0939

5. Conclusions

This study introduced a predictive method for a tool's RUL, utilizing joint feature selection and BiLSTM optimized by enhanced northern goshawk optimization. The efficacy of this novel methodology has been substantiated through meticulous milling experiments, culminating in several pertinent conclusions.

- (1) On an array of six benchmark test functions, the MSANGO algorithm exhibited exemplary convergence capabilities and robustness. This performance testifies to the MSANGO algorithm's adeptness at striking an effective balance between global and local search mechanisms, courtesy of its two innovative improvement strategies. This equilibrium enables the algorithm to exhibit superior convergence attributes in addressing optimization challenges.
- (2) The joint feature selection method introduced in this article concurrently accounts for both monotonicity and relevance in the selection of features; this method manifests its strengths in identifying features intimately linked with a tool's RUL. Such an approach offers invaluable insights and directives for the formulation of predictive models for tool RUL.
- (3) The integration of the MSANGO algorithm with BiLSTM notably surpasses the other three predictive models, exhibiting at least a 7.49% and 12.43% improvement in the MAE and RMSE metrics, respectively, across the three experimental groups, thereby refining the accuracy of predictions. Additionally, the MSANGO algorithm compared to the NGO algorithm reduces training time by at least 8%, further demonstrating its efficiency and effectiveness in real-world applications.

The proposed tool RUL prediction method provides an effective approach for predicting tool life in machining operations, demonstrating its practical application value; however, to enhance the method's applicability and accuracy, future studies might explore its use in more diverse operational settings. While the MSANGO algorithm has shown some potential in handling multimodal test functions, there is room for improvement in its stability. Future efforts could include the development of more consistent strategies or

optimization techniques to reduce the algorithm's variability, thus improving its stability and predictive precision.

Author Contributions: Conceptualization, J.W. (Jianwei Wu); Methodology, J.W. (Jianwei Wu); Software, J.W. (Jiaqi Wang); Validation, J.W. (Jianwei Wu) and J.W. (Jiaqi Wang); Formal analysis, J.W. (Jianwei Wu); Investigation, J.W. (Jianwei Wu); Resources, J.W. (Jianwei Wu); Data curation, J.W. (Jianwei Wu); Writing—original draft, J.W. (Jianwei Wu); Writing—review & editing, H.C.; Visualization, J.W. (Jianwei Wu); Supervision, H.C.; Project administration, H.C.; Funding acquisition, J.W. (Jianwei Wu) and H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of the People's Republic of China (grant number: 51975535) and a project supported by the Scientific Research Fund of Zhejiang Provincial Education Department (no. Y202353171).

Data Availability Statement: The data will be made available by the authors on request.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

Abbreviations

RUL	Remaining useful life
MVMD	Multivariate variational mode decomposition
IMF	Intrinsic mode function
UIC	Uniform information coefficient
MKT	Mann–Kendall trend test
LSTM	Long short-term memory
BiLSTM	Bidirectional long short-term memory
NGO	Northern goshawk optimization
MSA	Modified sine algorithm
MSANGO	Enhanced northern goshawk optimization
SCA	Sine cosine algorithm
WOA	Whale optimization algorithm
MAE	Mean absolute error
RMSE	Root mean square error
MAPE	Mean absolute percentage error

References

- Cai, W.L.; Zhang, W.J.; Hu, X.F.; Liu, Y.C. A hybrid information model based on long short-term memory network for tool condition monitoring. *J. Intell. Manuf.* **2020**, *31*, 1497–1510. [[CrossRef](#)]
- Warke, V.; Kumar, S.; Bongale, A.; Kamat, P.; Kotecha, K.; Selvachandran, G.; Abraham, A. Improving the useful life of tools using active vibration control through data-driven approaches: A systematic literature review. *Eng. Appl. Artif. Intell.* **2024**, *128*, 107367. [[CrossRef](#)]
- Karabacak, Y.E. Deep learning-based CNC milling tool wear stage estimation with multi-signal analysis. *Ekspluat. Niezawodn.* **2023**, *25*, 168082. [[CrossRef](#)]
- Zhang, X.Y.; Lu, X.; Li, W.D.; Wang, S. Prediction of the remaining useful life of cutting tool using the Hurst exponent and CNN-LSTM. *Int. J. Adv. Manuf. Technol.* **2021**, *112*, 2277–2299. [[CrossRef](#)]
- Sayyad, S.; Kumar, S.; Bongale, A.; Kotecha, K.; Abraham, A. Remaining Useful-Life Prediction of the Milling Cutting Tool Using Time-Frequency-Based Features and Deep Learning Models. *Sensors* **2023**, *23*, 5659. [[CrossRef](#)] [[PubMed](#)]
- Zhu, Q.S.; Sun, W.F.; Zhou, Y.Q.; Gao, C. A tool wear condition monitoring approach for end milling based on numerical simulation. *Ekspluat. Niezawodn.* **2021**, *23*, 371–380. [[CrossRef](#)]
- An, Q.L.; Tao, Z.R.; Xu, X.W.; El Mansori, M.; Chen, M. A data-driven model for milling tool remaining useful life prediction with convolutional and stacked LSTM network. *Measurement* **2020**, *154*, 107461. [[CrossRef](#)]
- Chacón, J.L.F.; de Barrera, T.F.; García, A.; de Buruaga, M.S.; Badiola, X.; Vicente, J. A Novel Machine Learning-Based Methodology for Tool Wear Prediction Using Acoustic Emission Signals. *Sensors* **2021**, *21*, 5984. [[CrossRef](#)]
- Cheng, Y.N.; Jin, Y.B.; Gai, X.Y.; Guan, R.; Lu, M.D. Prediction of tool wear in milling process based on BP neural network optimized by firefly algorithm. *Proc. Inst. Mech. Eng. Part E J. Process. Mech. Eng.* **2023**. [[CrossRef](#)]
- Chang, W.Y.; Hsu, B.Y. Tool life prediction via SMB-enabled monitor based on BPNN coupling algorithms for sustainable manufacturing. *Ai Edam* **2023**, *37*, e20. [[CrossRef](#)]
- Yu, W.W.; Huang, H.; Guo, R.L.; Yang, P.Q. Tool Wear Prediction Based on Attention Long Short-term Memory Network with Small Samples. *Sens. Mater.* **2023**, *35*, 2321–2335. [[CrossRef](#)]

12. Bazi, R.; Benkedjough, T.; Habbouche, H.; Rechak, S.; Zerhouni, N. A hybrid CNN-BiLSTM approach-based variational mode decomposition for tool wear monitoring. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 3803–3817. [[CrossRef](#)]
13. Singh, P.; Chaudhury, S.; Panigrahi, B.K. Hybrid MPSO-CNN: Multi-level Particle Swarm optimized hyperparameters of Convolutional Neural Network. *Swarm Evol. Comput.* **2021**, *63*, 100863. [[CrossRef](#)]
14. Li, X.W.; Qin, X.J.; Wu, J.X.; Yang, J.F.; Huang, Z.X. Tool wear prediction based on convolutional bidirectional LSTM model with improved particle swarm optimization. *Int. J. Adv. Manuf. Technol.* **2022**, *123*, 4025–4039. [[CrossRef](#)]
15. Yang, C.; Jiang, Y.T.; Liu, Y.; Liu, S.L.; Liu, F.P. A novel model for runoff prediction based on the ICEEMDAN-NGO-LSTM coupling. *Environ. Sci. Pollut. Res.* **2023**, *30*, 82179–82188. [[CrossRef](#)] [[PubMed](#)]
16. Zhong, J.R.; Chen, T.X.; Yi, L.H. Face expression recognition based on NGO-BiLSTM model. *Front. Neurobot.* **2023**, *17*, 1155038. [[CrossRef](#)] [[PubMed](#)]
17. Rehman, N.u.; Aftab, H. Multivariate variational mode decomposition. *IEEE Trans. Signal Process.* **2019**, *67*, 6039–6052. [[CrossRef](#)]
18. Mousavi, A.; Baraniuk, R.G. Uniform Partitioning of Data Grid for Association Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 1098–1107. [[CrossRef](#)] [[PubMed](#)]
19. Wei, X.D.; Yang, J.; Luo, P.P.; Lin, L.G.; Lin, K.L.; Guan, J.M. Assessment of the variation and influencing factors of vegetation NPP and carbon sink capacity under different natural conditions. *Ecol. Indic.* **2022**, *138*, 108834. [[CrossRef](#)]
20. Guo, F.Z.; Rasmussen, B. Predictive maintenance for residential air conditioning systems with smart thermostat data using modified Mann-Kendall tests. *Appl. Therm. Eng.* **2023**, *222*, 119955. [[CrossRef](#)]
21. Dehghani, M.; Hubálovsky, S.; Trojovský, P. Northern Goshawk Optimization: A New Swarm-Based Algorithm for Solving Optimization Problems. *IEEE Access* **2021**, *9*, 162059–162080. [[CrossRef](#)]
22. Ma, G.Y.; Yue, X.F. An improved whale optimization algorithm based on multilevel threshold image segmentation using the Otsu method. *Eng. Appl. Artif. Intell.* **2022**, *113*, 104960. [[CrossRef](#)]
23. Wang, Y.J.; Wang, G.G.; Tian, F.M.; Gong, D.W.; Pedrycz, W. Solving energy-efficient fuzzy hybrid flow-shop scheduling problem at a variable machine speed using an extended NSGA-II. *Eng. Appl. Artif. Intell.* **2023**, *121*, 105977. [[CrossRef](#)]
24. Luo, Y.B.; Dai, W.M.; Ti, Y.W. Improved sine algorithm for global optimization. *Expert Syst. Appl.* **2023**, *213*, 118831. [[CrossRef](#)]
25. Pelusi, D.; Mascella, R.; Tallini, L.; Nayak, J.; Naik, B.; Abraham, A. Neural network and fuzzy system for the tuning of Gravitational Search Algorithm parameters. *Expert Syst. Appl.* **2018**, *102*, 234–244. [[CrossRef](#)]
26. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
27. Zhang, C.; Ma, H.X.; Hua, L.; Sun, W.; Nazir, M.S.; Peng, T. An evolutionary deep learning model based on TVFEMD, improved sine cosine algorithm, CNN and BiLSTM for wind speed prediction. *Energy* **2022**, *254*, 124250. [[CrossRef](#)]
28. Yang, B.; Wang, Y.S.; Zhan, Y.D. Lithium Battery State-of-Charge Estimation Based on a Bayesian Optimization Bidirectional Long Short-Term Memory Neural Network. *Energies* **2022**, *15*, 4670. [[CrossRef](#)]
29. 2010 PHM Society Conference Data Challenge. 2010. Available online: <https://www.phmsociety.org/competition/phm/10> (accessed on 29 July 2024).
30. Karabacak, Y.E. Intelligent milling tool wear estimation based on machine learning algorithms. *J. Mech. Sci. Technol.* **2024**, *38*, 835–850. [[CrossRef](#)]
31. Mishra, D.; Awasthi, U.; Pattipati, K.R.; Bollas, G.M. Tool wear classification in precision machining using distance metrics and unsupervised machine learning. *J. Intell. Manuf.* **2023**, 1–25. [[CrossRef](#)]
32. She, C.X.; Li, K.X.; Ren, Y.H.; Li, W.; Shao, K. Tool wear prediction method based on bidirectional long short-term memory neural network of single crystal silicon micro-grinding. *Int. J. Adv. Manuf. Technol.* **2024**, *131*, 2641–2651. [[CrossRef](#)]
33. De Barrena, T.F.; Ferrando, J.L.; García, A.; Badiola, X.; de Buruaga, M.S.; Vicente, J. Tool remaining useful life prediction using bidirectional recurrent neural networks (BRNN). *Int. J. Adv. Manuf. Technol.* **2023**, *125*, 4027–4045. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.