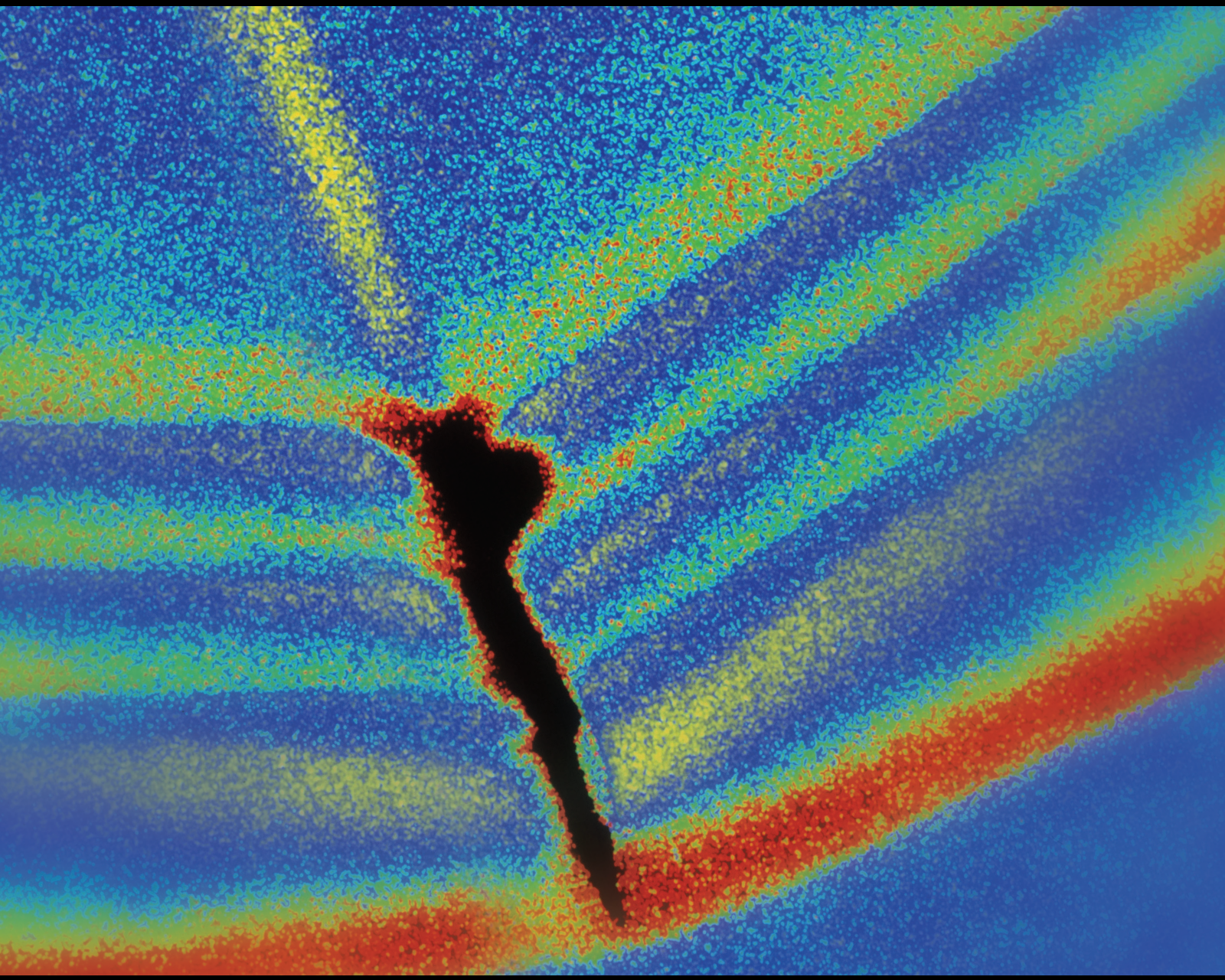
Shock and Vibration

Advances in Prognostics and Health Management for Intelligent Manufacturing

Lead Guest Editor: Tang-Bin Xia
Guest Editors: Xiaoning Jin, Rongxi Wang, Xi Gu, and Xiaolei Fang



Advances in Prognostics and Health Management for Intelligent Manufacturing

Shock and Vibration

Advances in Prognostics and Health Management for Intelligent Manufacturing

Lead Guest Editor: Tang-Bin Xia
Guest Editors: Xiaoning Jin, Rongxi Wang, Xi Gu, and Xiaolei Fang

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Shock and Vibration
Volume 2024, Article ID 9874938, 1 page
https://doi.org/10.1155/2024/9874938

Retraction

Retracted: Research on Optimizing Selection and Optimizing Matching Technologies of Aeroengine Fan Rotor Blades

Shock and Vibration

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] L. Li, K. Chen, J. Gao, Z. Gao, and J. Liu, "Research on Optimizing Selection and Optimizing Matching Technologies of Aeroengine Fan Rotor Blades," *Shock and Vibration*, vol. 2021, Article ID 5595535, 17 pages, 2021.

Hindawi
Shock and Vibration
Volume 2024, Article ID 9872819, 1 page
https://doi.org/10.1155/2024/9872819

Retraction

Retracted: Parallel-Machine Scheduling with DeJong's Learning Effect, Delivery Times, Rate-Modifying Activity, and Resource Allocation

Shock and Vibration

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] L. Sun, B. Wu, and L. Ning, "Parallel-Machine Scheduling with DeJong's Learning Effect, Delivery Times, Rate-Modifying Activity, and Resource Allocation," *Shock and Vibration*, vol. 2021, Article ID 6687525, 10 pages, 2021.

Hindawi
Shock and Vibration
Volume 2024, Article ID 9756979, 1 page
https://doi.org/10.1155/2024/9756979

Retraction

Retracted: Complex Behavior of Droplet Transfer and Spreading in Cold Metal Transfer

Shock and Vibration

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] S. Yang, Y. Xing, F. Yang, and J. Cao, "Complex Behavior of Droplet Transfer and Spreading in Cold Metal Transfer," *Shock and Vibration*, vol. 2020, Article ID 6650155, 11 pages, 2020.

Hindawi
Shock and Vibration
Volume 2023, Article ID 9795858, 1 page
https://doi.org/10.1155/2023/9795858

Retraction

Retracted: Research on Quality Anomaly Recognition Method Based on Optimized Probabilistic Neural Network

Shock and Vibration

Received 17 October 2023; Accepted 17 October 2023; Published 18 October 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] L. Li, K. Chen, J. Gao, and H. Li, "Research on Quality Anomaly Recognition Method Based on Optimized Probabilistic Neural Network," *Shock and Vibration*, vol. 2020, Article ID 6694732, 11 pages, 2020.

Hindawi
Shock and Vibration
Volume 2021, Article ID 9937846, 15 pages
https://doi.org/10.1155/2021/9937846

Research Article

Remaining Useful Life Estimation through Deep Learning Partial Differential Equation Models: A Framework for Degradation Dynamics Interpretation Using Latent Variables

Sergio Cofre-Martel,[1] **Enrique Lopez Droguett**,[1,2] **and Mohammad Modarres**[1]

[1]*Department of Mechanical Engineering, Center of Risk and Reliability, University of Maryland, College Park, MD, USA*
[2]*Department of Civil and Environmental Engineering & The Garrick Institute for the Risk Sciences, University of California, Los Angeles, USA*

Correspondence should be addressed to Sergio Cofre-Martel; scofre@umd.edu

Received 19 March 2021; Accepted 19 May 2021; Published 27 May 2021

Academic Editor: Tangbin Xia

Remaining useful life (RUL) estimation is one of the main objectives of prognostics and health management (PHM) frameworks. For the past decade, researchers have explored the application of deep learning (DL) regression algorithms to predict the system's health state behavior based on sensor readings from the monitoring system. Although the state-of-art results have been achieved in benchmark problems, most DL-PHM algorithms are treated as black-box functions, giving little-to-no control over data interpretation. This becomes an issue when the models unknowingly break the governing laws of physics when no constraints are imposed. The latest research efforts have focused on applying complex DL models to achieve low prediction errors rather than studying how they interpret the data's behavior and the system itself. This paper proposes an open-box approach using a deep neural network framework to explore the physics of a complex system's degradation through partial differential equations (PDEs). This proposed framework is an attempt to bridge the gap between statistic-based PHM and physics-based PHM. The framework has three stages, and it aims to discover the health state of the system through a latent variable while still providing a RUL estimation. Results show that the latent variable can capture the failure modes of the system. A latent space representation can also be used as a health state estimator through a random forest classifier with up to a 90% performance on new unseen data.

1. Introduction

As the evolution of traditional condition-based maintenance (CBM) techniques, prognostics and health management (PHM) frameworks seek to study and predict the evolution of a system's health state based on data collected from sensor readings. This data is expected to contain critical information related to the system's past and current health state [1]. The main goal of a PHM framework is to estimate the remaining useful life (RUL) of the system, which is later used as a metric for decision-making during the optimization of maintenance policies and health management [1, 2]. Obtaining accurate RUL estimations from sensor data requires a precise knowledge and understanding of the system and, depending on the available information, three main approaches can be implemented for the RUL estimation: physics-based models (PBMs) [3], data-driven approaches (DDAs) [4], and hybrid methods [5]. In this context, we present a deep learning framework to uncover the physics of complex systems' degradation. The framework is inspired by physics-informed neural networks and can be considered a hybrid method for the health state assessment and RUL estimation.

Hybrid methods combine PBMs and DDAs to overcome their weaknesses and combine their strengths [5, 6]. On the one hand, PBMs rely on a mathematical representation to describe the degradation physics governing the system. These methods require a few data points for the training process and yield results directly interpretable by the user. Although PBMs are highly accurate and reliable, they are

system-dependent models and cannot be easily scaled and adapted from one system to another. This is why PBMs reliability prognostics studies are usually limited to local crack propagation and corrosion [3], making their direct application to complex systems a challenging task.

On the other hand, machine learning [7] and deep learning (DL) [8] have become the preferred application of DDAs to PHM. These techniques provide an alternative to analyze complex systems when the physics behind the degradation process is unknown. These can extract abstract information and features from massive datasets while building and discovering complex functional and temporal relationships from the data [9]. Deep learning approaches have been implemented in a great variety of systems for prognostics purposes, such as lithium-ion batteries state of health (SOH) and state of charge (SOC) estimation, [10–13], RUL estimation in rolling bearings [14–16], and turbofan engines [17–20].

Although great advances have been made in DL applications to PHM, there are still many challenges to face before implementing these models in the industry [2, 9]. One of these challenges is model interpretability, as DL applications create explainable models that cannot be directly interpreted by the end-user. This has had a detrimental effect on the engineers' trust to implement DL models in real-life systems [21]. Without interpretability, one can only rely on performance metrics to select a model. This can bias the user to choose models with a low error on their training and validation data, regardless of the model's true representation of the system under study. In this regard, third-party software and packages have been developed, providing information on feature relevance for models' predictions [22, 23]. For instance, in [21], the authors presented an algorithm called Local Interpretable Model-Agnostic Explanations (LIME) that provides insight into the relevance that input features have on an ML classification model's prediction. A similar framework was presented by Lundberg and Lee [22] called Shapley Additive explanations (SHAP) for deep learning models. This framework assigns weight values to the input features as importance measures of their effect on the DL model's output. These third-party algorithms provide valuable information for the models' interpretability: nevertheless, they primarily address classification models focusing on natural language or image processing and cannot be implemented within the model itself. Such algorithms can be used as preprocessing or postprocessing techniques. However, they do not influence the model's performance as feature relevance does not have any influence on the models' learning process.

In the context of DL-PHM models, two elements heighten the barriers for model interpretability that are yet to be addressed: the use of time as an explicit variable and the explicit relationship between the physics of the system and the input variables of the model. Indeed, most of the DL-PHM models do not explicitly consider time as a variable in their calculations. Works that apply recurrent neural networks (RNN) and its long-short term memory (LSTM) variation [24–26] use input data with time implicitly embedded through consecutive feature logs, which are then interpreted by the model. Here, the network is trained with a sequence of data points to understand the time scale represented in the data. Thus, the network is given the additional task of interpreting the time relationship among its features. However, new unseen data logs might have different temporal behavior in their log sequences. Likewise, embedding the physics of degradation of a system to a DL framework is a challenging task. Although advances have been made in this area [27, 28], solutions heavily rely on the availability of an empirically based mathematical model (i.e., crack propagation and corrosion, resp.) to describe the damage propagation or future behavior of the system degradation.

The latest advances in DL algorithms have shown that it is possible to embed partial differential equations (PDEs) to DL models. Raissi et al. [29] presented a physics-informed neural network (PINN) framework to solve PDEs by incorporating them as a penalization term to the cost function during the neural network (NN) training process. The framework also allows us to discover PDEs embedded in the data when an explicit equation is not available. This opens the door to create a dynamic relationship between the sensor data and the degradation process in complex systems using DL models in PHM. In this paper, we present a deep neural network (DNN) framework for RUL prognostics that maps the monitoring data and time to a latent variable representation linked to the system's degradation dynamics through a PDE-like penalization function. Once the model is trained, the latent space representation works as a system health estimator quantitatively and qualitatively. In other words, this framework resembles a PDE, where, given initial feature values (i.e., initial conditions), the algorithm can estimate a RUL value through the PDE solution for a given time after the given initial conditions.

Up to date, most DL applications to PHM focus on either diagnostics or prognostics. Very few research works have provided frameworks that can perform these two tasks simultaneously. For instance, Kim and Sohn [30] presented a multitask deep CNN with double outputs, one for prognostics and another for diagnostics. This requires manually handcrafting labels and significantly increases the number of trainable parameters. The training of RNN models requires input data shaped as time windows, which can be impractical to create when sensor data is not sampled at a constant rate or contains missing data points, which is common in real case scenarios. Time windows can also be a source of overfitting if the preprocessing of the data is not carefully done. Further, none of the aforementioned frameworks provide interpretation or visualization of their results. As such, the contributions of this paper are the following:

(1) We present a framework that aims to bridge the gap between statistics-based and physics-based PHM applications.

(2) Inspired in PINN, the proposed framework uses a dynamic PDE-like penalization function that explicitly binds the monitoring data and time to the system's degradation process. This is the first application of PINN to DL-PHM frameworks to the authors' best knowledge.

(3) By using time explicitly, the framework is able to capture the temporal behavior of the data directly. This differs from other commonly used DL algorithms in PHM frameworks such as convolutional neural networks (CNNs) and long short-term memory (LSTM) neural networks which infer these relationships from the data structure instead.

(4) The framework delivers a combined diagnostics and prognostics analysis of a system by providing a RUL estimation along with a health classifier between the system's healthy and degraded states.

(5) The proposed framework also provides interpretability of the system's health state through the visual representation of a latent variable.

The remainder of this paper is structured as follows: Section 2 presents the background behind PDEs applied to DL. Section 3 discusses the proposed DL framework, which is trained with the dataset presented in Section 4. The obtained results and their discussion are presented in Section 5. Section 6 outlines the main conclusions and remarks of this study.

2. Physics-Informed Deep Learning

Most DL algorithms' applications are implemented as black-box functions in which the extraction of abstract relationships in the data is left for the machine to find. In this regard, efforts have been made to provide both interpretation and constraints to these techniques from a physics perspective. Raissi et al. [29] proposed a physics-informed neural network framework that integrates and solves PDEs given a set of initial and boundary conditions. In this work, the authors show that a PINN framework can also be used to recover or create PDEs from the data itself without any prior underlying knowledge on the physics governing the system under study. To understand how this algorithm works, it is necessary to quickly review the architecture behind DL models as function representations and the principles of PDEs.

The main structure in DL is deep neural networks. Here, an input value is evaluated through sequential combinations of nonlinear functions to yield the desired output value. Hence, one can represent the output y of a NN as a function in the form of

$$\widehat{y} = f(X, W), \tag{1}$$

where $f(X, W)$ is the NN, X are the input values, and W is a tensor of parameters called weights, which defines the function. Two key components compose a NN: layers and hidden units (also known as neurons). A layer is a nonlinear function of an input value, commonly represented as

$$h_i = \sigma\left(W_i^T h_{i-1} + b_i\right), \tag{2}$$

where h_i is the hidden layer i, represented by its weight matrix W_i and bias vector b_i. Notice that the relationship among h_i, W_i, and X is a simple linear regression. This is then evaluated in a nonlinear function σ, also referred to as activation function. The dimensions of the weight matrix for a NN layer are determined by the number of neurons from the previous layer and its number of neurons. As it can be observed in equation (2), a layer takes as input the output of the previous layer, and it yields an output, which then goes on into the next hidden layer, and so on until the output layer is reached. For instance, equation (3) shows a two-layer NN of input X, output $\widehat{y}$, and activation function σ:

$$\widehat{y} = \sigma\left(W_2^T \sigma\left(W_1^T X + b_1\right) + b_2\right) = f(X, W). \tag{3}$$

Thus, for a given dataset (X, y), the parameters defining the NN in equation (3) are optimized to minimize the average of the squared errors, which is the so-called loss function described in equation (4). Given a set of data points (often referred to as dataset), equation (4) can be optimized using gradient descent [31] and backpropagation [32]:

$$\text{loss} = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - y_i\right)^2. \tag{4}$$

On the other hand, PDEs model the behavior of a function of interest based on the relationship between its partial derivatives with respect to its input variables. For instance, let $u(z, t)$ be a two-dimensional function of space and time. Then, a PDE for $u(z, t)$ can be represented as

$$u_t = F\left(z, u, u_z, \ldots\right), \tag{5}$$

where subindexes indicate partial derivatives of the function $u(z, t)$, for example, $u_z = \partial u(z, t)/\partial z$. The right-hand side of equation (5) is represented by a function F with input variables related to the space variable. In their proposed methodology for PINNs, Raissi et al. took advantage of automatic differentiation [33] to formulate a PDE-like penalization function by considering the target variable of interest by $u(z, t)$ (e.g., velocity field, temperature) as the output of a NN that takes z and t as input variables. As such, one can use automatic differentiation to calculate the exact derivative of the NN representing u with respect to any of its input variables (e.g., u_z, u_t). This allows the creation of a PDE in the form of equation (5), where u_t is the time derivative of the output variable. The function F on the right-hand side is represented by a second NN, which takes the spatial variables and their corresponding u derivatives as input variables. Equation (5) can be written as a cost function in terms of a function f described in equation (6). Adding f as a penalization term to the training cost function of the DL model would then bind the behavior of the parameters representing the NNs of $u(z, t)$ and F through the PDE. Note that if the right-hand side function $F(z, u, u_z, \ldots)$ in equation (5) is known, we can directly implement it to the training cost function. Hence, the optimization cost function of the neural network can be written as shown in equation (7), as the sum of the loss function in equation (4), and the square of f in equation (6). Here, λ is the weight value for the penalization function, and M is the number of points to be tested in the PDE. These can be collocation points, initial conditions, or boundary conditions:

$$f u_t - NN\left(z, u, u_z, \ldots\right) = 0. \tag{6}$$

$$\text{Cost} = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - y_i\right)^2 + \lambda \frac{1}{M} \sum_{j=1}^{M} f^2. \tag{7}$$

One of the first implementations of NNs to approximate PDEs was presented in [34], focused on the numerical challenges presented by nonlinear PDEs on continuous mechanical systems. Here, the output of a DNN was used as an approximation for the solution of the PDE (i.e., u), and an unconstrained optimization function was enforced at specific layers and neurons of the network. This architecture is used to solve a linear Poisson equation and thermal conduction with a nonlinear heat generation problem. Later research showed applications of DNNs to solve general coupled PDEs based on Dirichlet and Newman boundary conditions [35]. These first studies mostly focused on the computational efficiency of using NNs to solve PDEs when compared to traditional methods such as finite-element analysis. However, at the time, studies were limited by computational hardware capabilities. Given the nature of their definition, PINNs have mostly been applied in the fluid dynamic research community. The case study presented by Raissi et al. [29] uses Burger's equation for three possible applications: (1) solve a known PDE given initial and boundary conditions, (2) find parameters that govern a known PDE based on data from the objective space, and (3) find and solve an unknown PDE solely based on data from the objective space.

There are currently no PINN applications to PHM frameworks in the reliability community. This is mainly due to the lack of equations that can link a complex system's degradation dynamics with its condition monitoring sensor readings. Nevertheless, most DL-PHM frameworks seek to relate the monitoring variables with the system's diagnosis and prognosis. As such, the PINN approach proposed in [29] presents an opportunity to seek and find possible unknown PDEs that can relate sensor measurements to the system's degradation process.

3. Proposed Framework

Obtaining models that simultaneously yield an interpretable health estimator and traditional prognostics metrics is an ongoing challenge in DL-PHM models [2, 9]. An interpretable model allows the user to trust its prediction, which is critical for implementing DL-PHM models for the health management of real systems. Training a DNN to represent the degradation process in a complex system is difficult due to the lack of mathematical models to describe its physics of degradation. Moreover, most DL models applied to PHM do not consider time as an input variable of the network. Thus, information regarding the degradation dynamics of the system is lost during the training process if not explicitly stated (as in PBMs). In the case of RUL estimation, another challenge is presented when creating labels for supervised models. Here, it is common to define a point at which the degradation process starts. This can be either at a fixed time before failure [34] or when a specific performance variable surpasses a predefined threshold value [35]. Both approaches impose a strict constraint to the RUL labels by assuming that the machinery under study will continue to operate in the same condition until its failure. A DL model trained with these labels will inevitably be biased towards this behavior, making it susceptible to errors when tested with new data. Nevertheless, we can overcome this uncertainty by giving interpretability to our model.

Since there are no available equations to directly map the health state of a complex system to its operational conditions, we propose a DNN framework to explore the degradation physics of a system through a latent space representation. The supervised framework is aimed at PHM prediction tasks, where operational data is available from the monitoring of a system. The framework establishes a relationship between a latent variable and a prognosis output variable through a PDE-like penalization function (equation (8)). By training the DNN to understand the dynamics of the degradation process, it is expected that the model will improve its generalization capabilities. Indeed, adding a PDE-like penalization to the loss function of the model creates a relationship between the input features of the model and the derivatives of the output value with respect to its independent variables. This effect can be boosted if the framework is given time as an input feature, rather than implicitly extracting it from a sequence. For metrics such as the RUL, the penalization function adds information on the degradation rate by considering temporal derivatives.

Figure 1 illustrates the proposed DNN framework. It yields RUL estimations through three stages, represented by three different NNs. The first stage maps the operational conditions (OCs) and the time t to a (possibly multidimensional) latent variable x. A second NN then takes both t and x to yield the RUL estimation of the system. A third NN is used to model the right-hand side of equation (5) $F(z, u, u_z, \ldots)$, which models the RUL's time derivative through a NN. This is the so-called dynamics of the PDE. The NN for each stage of the proposed framework is structured as follows:

(1) x-NN: the network takes the OCs and time as input variables and it outputs the latent variable x. It is comprised of 5 hidden layers of 3 units each and two units as an output layer. This accounts for 104 trainable parameters. Hyperbolic tangent (tanh) is used as the activation function. The dimensionality of the latent variable is a hyperparameter that needs to be tuned according to the system under study.

(2) RUL-NN: it takes both the latent variable x and time as input values and outputs the RUL of the system. It comprises of 5 hidden layers of 10 units each to yield one output unit with tanh as the activation function. The network encompasses 481 trainable parameters.

(3) Dynamics-NN: it takes the latent variable x and the derivatives $\mathrm{d}x/\mathrm{d}t$ and $\mathrm{dRUL}/\mathrm{d}x$ as input values. It outputs a function N that represents the dynamics of the system. This goes into the PDE-like penalization function. The network is comprised of 5 hidden layers of 10 units each. One output unit and a rectified linear unit (ReLU) as the activation function are used. This network also contains 481 trainable parameters.

With automatic differentiation, we can take the time derivative from the first and second stage of NNs. These are then combined to form a PDE-like penalization term, as shown in Figure 1. The penalization includes the time derivative from the RUL, which is related to the Dynamics-NN in a PDE-like form as shown in equation (8), where $\mathrm{d}(\mathrm{RUL})/\mathrm{d}t$ is the time derivative of the second stage NN, and

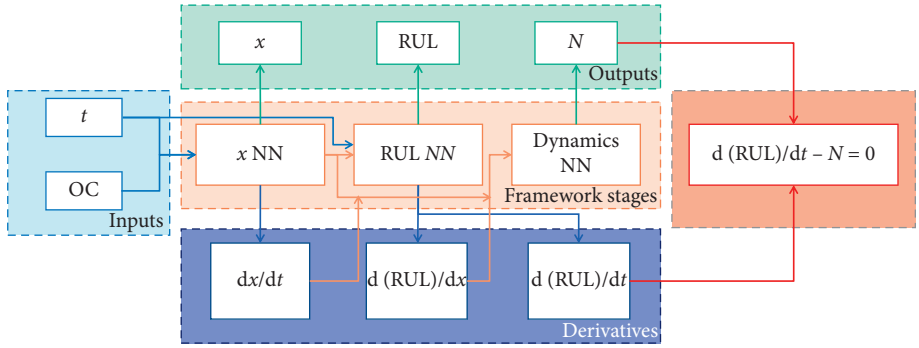


Figure 1: Proposed deep learning framework using a latent variable and a PDE-like penalization function.

N is the output from the third stage NN. The training cost function is then defined as shown in equation (9), where RUL_i and RUL_i are the objective and predicted RUL values, respectively:

$$f: \frac{\mathrm{d}(\mathrm{RUL})}{\mathrm{d}t} - N\left(x, \frac{\mathrm{d}x}{\mathrm{d}t}, \frac{\mathrm{d}(\mathrm{RUL})}{\mathrm{d}x}\right), \tag{8}$$

$$\mathrm{Cost} = \frac{1}{N}\sum_{i=1}^{N}\left(\mathrm{RUL}_i - \mathrm{RUL}_i\right)^2 + \lambda\frac{1}{N}\sum_{j=1}^{N} f^2. \tag{9}$$

The penalization function f thus creates a dynamical relationship between the RUL and the latent variable x, which in turn is related to the initial operating conditions and the time at which the RUL is evaluated. The framework is comprised of 1,066 trainable parameters, which is a low number when compared to other significantly more complex DL architectures for RUL estimation [36]. Having a model with fewer parameters to train prevents overfitting and reduces the training time, which can eventually facilitate its online implementation without the need for specialized hardware.

The proposed framework addresses many of the drawbacks mentioned above in DL applications for PHM. First, the network takes time as an input variable, along with the operational conditions of the system. The OCs represent the initial conditions for a PDE, and t corresponds to the point in the future at which it is desired to obtain the RUL value. In other words, for $t = 0$, the network behaves as most DL methods. That is, RUL is predicted based on the current OCs. Secondly, the use of a latent variable provides multiple advantages for both the training of the model and the later interpretation of its results:

(1) Dimensionality reduction: the usage of a latent variable helps capture and highlight important information related to the degradation process from the OCs. The dimensions of the latent variable dictate the number of dimensions that we can use for visualization purposes. In turn, visualizing a latent space provides additional tools to make an informed decision based on the model's output.

(2) Input variables for Dynamic-NN: the right-hand side function in equation (6) could take every possible derivative from the input OC values. The use of a latent variable reduces the number of derivatives fed into the Dynamics-NN, thus reducing the number of parameters of the network and its training time.

(3) Eliminate redundancy and noise from the data: due to the potentially high correlation among monitoring variables, it is common to observe that a lower-dimensional space can represent a system. This is the basic concept behind every data-driven approach for regression in PHM. Further, DNNs are known to remove noise levels in the input signals.

Note that out of the three stages, only the RUL-NN requires labels for the training process, since the latent variable x comes as a secondary outcome from the back-propagation training of the RUL-NN. On the other hand, the Dynamics-NN is trained solely from the penalization PDE term, which does not require any labels. Furthermore, if a degradation equation is available, for example, Paris' Law for crack propagation, it can be directly replaced for the Dynamics-NN, giving our proposed model flexibility according to the available information on the system under study.

To train models based on the proposed framework, the following steps must be followed:

(1) Preprocess the dataset. The input data to the framework has two essential elements: sensor measurements and prediction time horizon. Details on the dataset preparation are presented in Section 4. Given that this is a supervised framework, objective labels associated with the input values must also be provided.

(2) Define and set up the framework (Figure 1) according to the available data and information on the system under study. If available, a PBM (e.g., Paris' Law) can be included in the penalization function, replacing the Dynamic-NN. Otherwise, the Dynamic-NN is used to discover the system's degradation dynamics. Other hyperparameters such as the dimensionality of the latent variable x, the number of neurons and layers of each NN, and the penalty weight λ need to be selected as discussed in Section 5.

(3) Train the model based on the chosen framework with the preprocessed dataset. All NN stages within the

framework are trained simultaneously according to equation (9).

(4) Once the model is trained and depending on the selected dimensionality of the latent variable x, results can be visualized by evaluating new input values and plotting each dimension of x on a different axis. Here, the output values of the model (e.g., RUL) are used as a color map. This visualization allows directly assessing the relationship between the trained latent variable and the objective function. As discussed in Section 5, the latent variable can be used as a health estimator in the PHM context through a ML classifier.

4. Case Study: Dataset and Hardware

The proposed framework is tested using the benchmark dataset C-MAPSS due to the multiple research reports that have applied DL networks for its RUL estimation [6, 20, 24, 37, 38]. A detailed description of this dataset and its processing can be found in [39, 40]. This study's objective is not to improve the state-of-the-art results on this dataset regarding RUL estimation precision, but rather to introduce a new tool for PHM-DL models. Hence, only the FD001 and FD004 subdatasets from the C-MAPSS will be covered in detail. The dataset consists of 27 sensor variables for simulated engine runs. The FD001 dataset presents one failure mode and one operational condition. FD004 on the other hand presents two different failure modes, and it operates at six different conditions. The information on which failure mode caused the failure of the engine run is not provided with the dataset, nor are the conditions at which they were operating before the failure. Operational sensor readings are recorded for each cycle during an engine run. Each engine run starts at a random initial degradation level from which the engine operates until its failure.

As has been shown in past studies [40], only 14 out of the 27 sensors are statistically significant to model the RUL of the system, and thus these are the ones used for this study. Since the proposed framework is based on vanilla DNNs, there is no need to create time windows for the input data. However, we need to create a temporal dimension (i.e., feature) in order to train the proposed model. As such, the original dataset needs the following additional processing steps:

(1) Select all data logs for one engine run, from its initial starting point until its failure.

(2) For each operational cycle, add a column with an integer time t from 0 to 30 cycles.

(3) Create a label for the above operation data and time, which corresponds to the RUL value at time t since the initial point.

For instance, let us consider Engine 1, which contains a total of 192 log entries. If cycle number 100 is selected as the initial point, then, for $t = 0$, its corresponding label is RUL = 92; then, for $t = 1$, its label is RUL = 91; and so on until $t = 30$ is reached or until RUL = 1 (i.e., the engine fails). This process is repeated for each log entry of each engine, which increases the size of the original dataset. For instance, the FD001 subdataset size increases from 20,631 to 593,061 points. The input values are normalized using a MinMax scaler, which is a common practice when training DNNs [41]. Models are trained on Python 3.6 with the use of Tensorflow 2.0 and Keras. Windows is used as the operating system. The computer hardware consists of an Intel i7-9700k CPU, 32 GB of RAM memory, and a 24 GB Titan RTX GPU. The average training time in this machine is 140 seconds, while the evaluation time for new data entries is 0.01 seconds.

The value range of the newly added time feature column is an additional hyperparameter of the proposed framework. This will depend on the specific system under study, and in this case, it was selected based on the following reasoning. The time horizon for RUL estimation needs to be realistic. In this regard, if a system begins operating from an almost perfect health state, there would not be an indication of the degradation process within the monitoring data. Hence, it would be optimistic to expect the model to accurately estimate future RUL values at times close to the end of the system's life based on this data. As such, we should not train the model to yield RUL predictions at times exceeding the training RUL labels values. Since the RUL labels for the C-MAPSS range from 1 to 125 cycles, the time dimension should at most range from 0 to 125 cycles. Based on this reasoning, we tested the framework with prediction horizons from 0 to 100 cycles. We observed that the model's performance decreases significantly for horizons greater than 30 cycles. Thus, we chose this as the upper time limit, which accounts for almost one-quarter of the training label range.

5. Results and Discussion

We train the proposed framework for the FD001 and FD004 subdataset from C-MAPSS. Models are trained using 75% of the data randomly selected from the training set, with the remaining 25% left as a validation set. The test sets are provided separately [40]. NAdam optimizer [42] is used for the training process. The proposed framework comprises multiple hyperparameters; three of these have the most significant impact on the model's performance after training: the latent variable dimensionality, the penalty weight λ assigned to the PDE regularization function, and the number of training epochs. Figures 2 and 3 illustrate the results for the sensibility analysis of these three hyperparameters. For each combination of hyperparameters, 10-fold cross-validation was performed with random initial parameters. We compare the average cost function value on the training and validation set from the cross-validation process in these figures. The minimum cost is indicated with a red dashed horizontal line for each case.

Figure 2 shows the joint sensibility analysis for the number of training epochs and the latent variable dimension. On one hand, most cost values decrease with a higher number of training epochs for both the training and the validation set as expected. This behavior is shown by both subdatasets, independent of their complexity. On the other

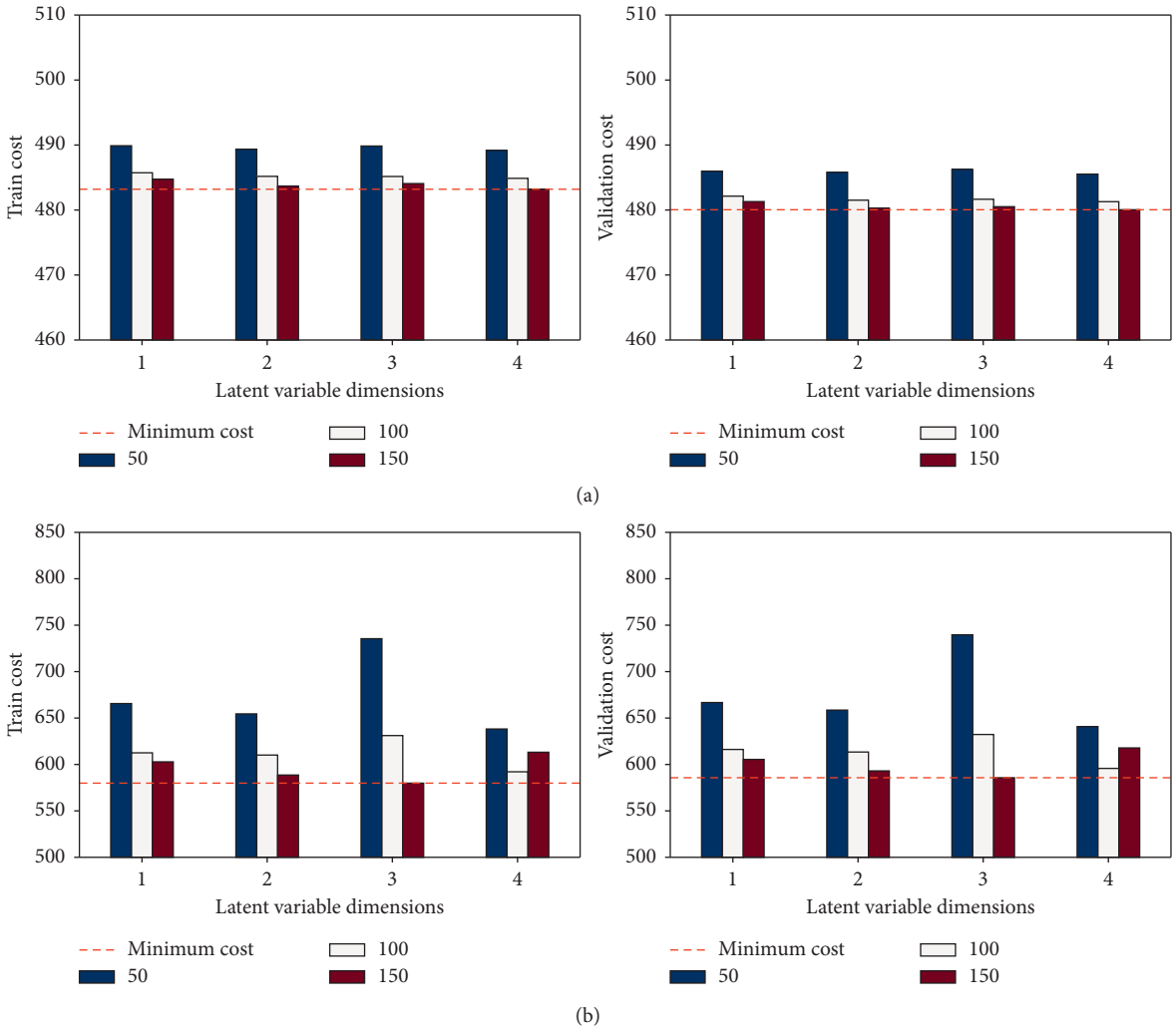


Figure 2: Sensibility analysis of the cost function by the number of training epochs. (a) FD001. (b) FD004.

hand, we can see that the best results are achieved with a two-dimensional latent variable on the FD001 set, while the FD004 set performs better with a three-dimensional latent variable. These results are consistent with the complexity difference between the datasets since the FD004 set contains six operational conditions and two failure modes. Thus, the model requires a higher latent variable dimensionality to represent the degradation process. Further, the results shown in Figure 2 for the FD001 set indicate that models have similar performance for a latent variable with more than two dimensions. In the case of the FD004 set, a similar performance is obtained for two and three dimensions.

Figure 3 presents the joint sensibility analysis for the PDE penalization weight value and latent variable dimension. The penalization function improves the generalization capabilities of the model, resulting in similar cost performance when evaluating the training and validation sets. However, the specific value of the weight penalization is the most difficult hyperparameter to analyze. A higher penalization value results in a more constrained model, and thus, its performance worsens when evaluating the training set. For instance, we can observe that, with a low penalization value, the model presents underfitting (i.e., the validation cost value is lower than the training) on the FD001 set regardless of the latent variable dimensionality. Nevertheless, a higher penalty value during the training process would give higher importance to the connection between the latent variable representation and the RUL of the system. This would explain the more consistent behavior between the training and validation set for the more complex FD004 set. Both datasets have a consistent cost value with higher penalization weights in the case of a two-dimension latent variable, particularly the FD004 set, where there is neither significant underfitting nor overfitting. Hence, a two-dimensional latent variable is better when considering the PDE penalization function.

From this hyperparameter analysis, a two-dimensional latent variable is selected due to its more consistent results. This is also a good choice for visualization purposes, given that one will be able to map all the dimensions in a two-dimensional latent space representation once the model is trained. Moreover, models are trained for 150 epochs and a

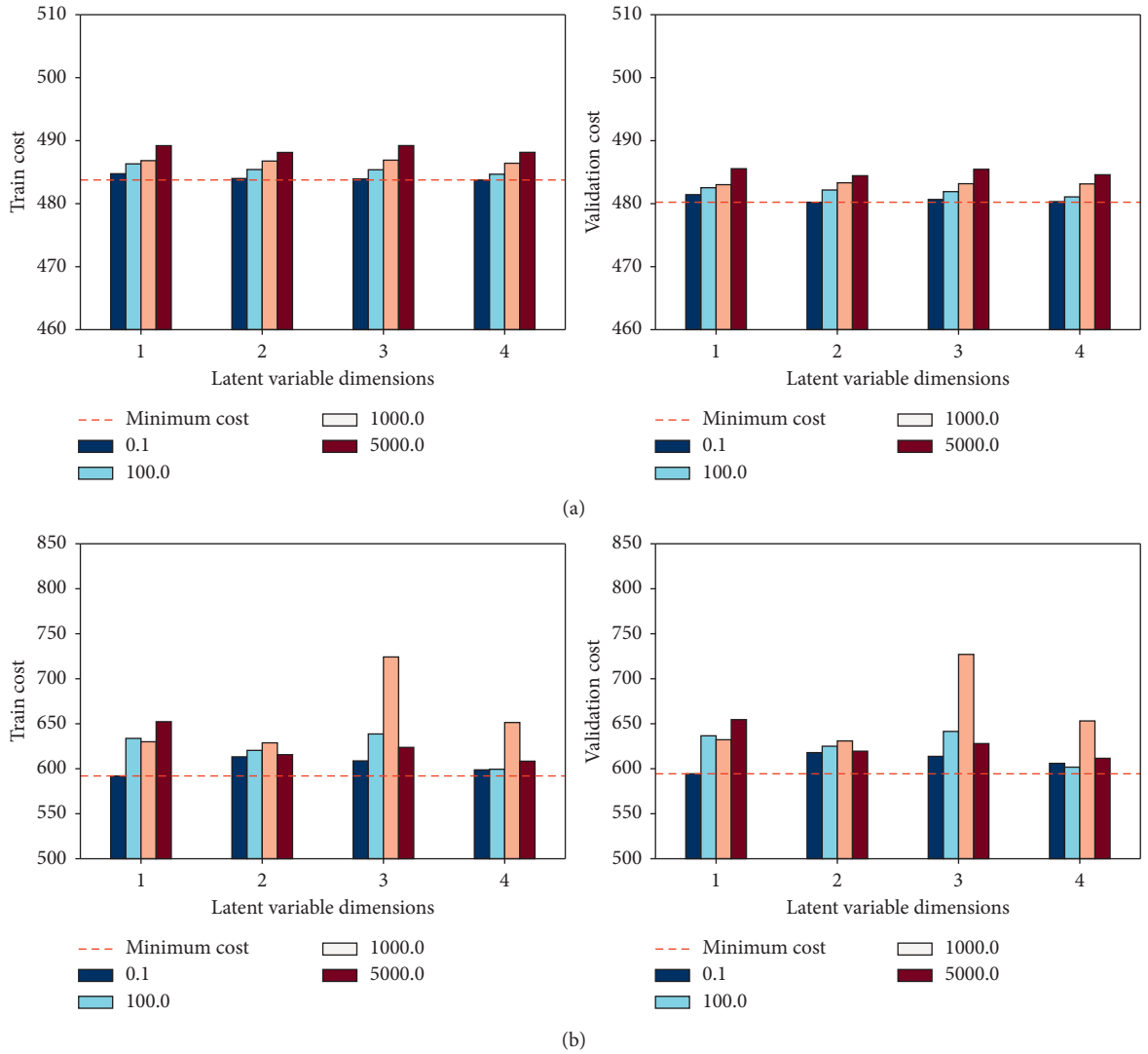


Figure 3: Sensibility analysis of the cost function by PDE penalty weight value λ. (a) FD001. (b) FD004.

penalty weight value of 100. Ten different models are trained for each dataset, each starting from random initial weights for the three NN stages.

Table 1 presents the results for the average root mean squared error (RMSE) obtained with the trained models for each dataset. FD001 models average an RMSE value of 17.14 cycles for its test set, while models for FD004 yield an average of 25.58 cycles. Figure 4 illustrates the training and validation cost throughout the training process. Here, it can be observed that both curves present an identical behavior. Also, these converge to the same cost value and, thus, the trained models have good generalization capabilities. We can attribute this behavior to the PDE penalization function added to the model. The dynamical relationship built between the latent variable and the RUL, as well as the inclusion of the time dimension, provides extra information on the degradation dynamics to the model. In turn, the model can yield consistent predictions for new unseen data. The behavior of the cost function during the training process is also consistent with the hyperparameters effects studied in Figures 2 and 3.

Table 1: Training and testing RUL RMSE values for models trained based on the proposed framework.

	Training RMSE	Test RMSE
FD001	21.96	17.14
FD004	24.72	25.58

Although the obtained RMSE values for the test sets are not as low as those obtained through other far more complex architectures, these are within the acceptable range for this case study [20]. Such complex architectures involve a high number of trainable parameters without providing interpretability tools for the end-user. For instance, a deep convolutional neural network (DCNN) for RUL estimation with over 180k trainable parameters was presented in [20]. This number increases further when additional layers of analysis are added to CNNs, such as multiscale blocks [43] or bidirectional LSTMs [44]. They require more preprocessing for their training data and can present overfitting while requiring specialized hardware for a fast training process.

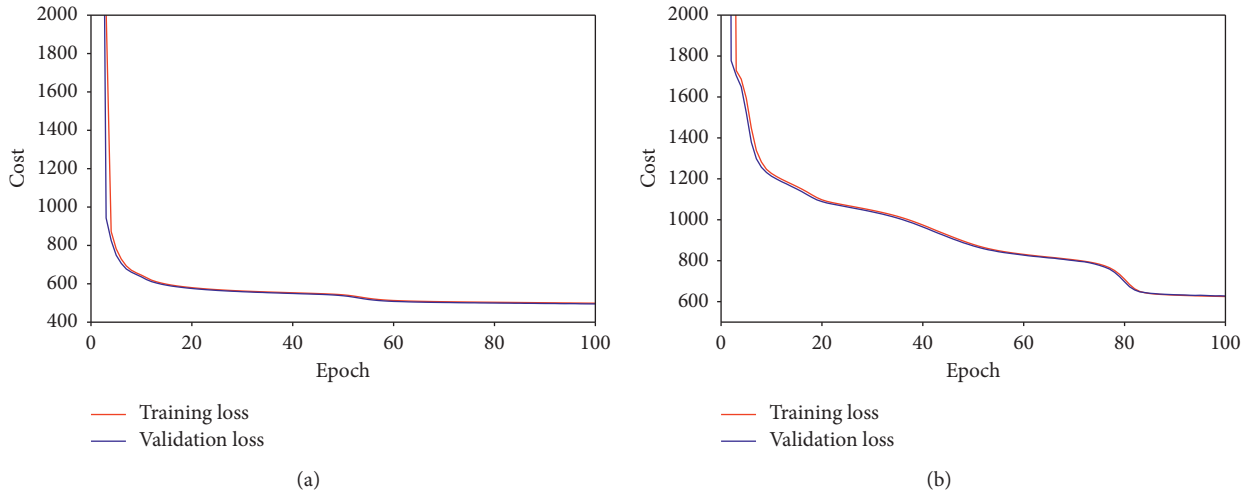


Figure 4: Training and validation cost value per number of epochs during the training process for the FD001 (a) and FD004 (b) subdatasets.

Additionally, the more trainable parameters a model has, the more training data is needed to prevent overfitting.

The proposed framework's most important output is the latent space representation of the trained models. Indeed, Figure 5 illustrates the predicted RUL values mapped to their corresponding estimated latent variable space for both the training and test sets. Both dimensions from the latent variable x (i.e., x_1, x_2) are plotted with their corresponding RUL values represented as a color map. Figure 5 shows three different RUL mappings for each subdataset. On the left, the RUL training labels are mapped to their corresponding latent space representation. At the center, RUL values predicted by the trained model evaluated with the training set (i.e., same input data as the previous case) are mapped to their corresponding latent space. On the right, similar to the first figure, an RUL colormap is presented for the latent space representation of the test set labels.

Results presented in Figure 5 for the training set show that the trained model smooths the RUL value representation to the latent space. This creates a continuous relationship between the operational conditions and the RUL of the system. Given the linear relationship between the RUL and time, a health index related to the RUL is analogous to an indicator of the temporal evolution of the system's degradation process. Hence, we can consider the latent space representation in Figure 5 as a health state indicator related to the system's underlying degradation process. Moreover, Figure 5 shows that both subdatasets present different shapes on their latent space representation. This is expected since both datasets present a different number of failure modes. Indeed, given that the FD001 set has only one failure mode, a latent space domain following a straight-line path from low to high RUL values makes us think this is a good representation of the degradation process of this particular system. This degradation path is also simpler than its FD004 counterpart. In the latter, we see that, from a healthy state (i.e., high RUL values), the latent space presents a bifurcation into two degradation paths. Since this dataset comes from a system with two different failure modes, we believe these degradation paths can be the model's interpretation of the failure modes. Unfortunately, information on which failure mode caused the system's failures is not available to confirm this observation.

In the case of the test set representation from each subdataset in Figure 5(a), we observe that both the RUL mapping and the shape of the latent space representation of the test set are consistent with those obtained for the training set (center images). This reinforces the generalization capabilities of the models discussed in Figure 4, where we observe that the training and validation cost curves were almost identical throughout the training process. These results from the test sets indicate that the latent space can indeed be used as an indicator of the system's health state. This interpretability is why including time as an input variable becomes crucial to our proposed framework. By including time, it is possible to obtain the temporal derivative of the RUL (i.e., the RUL dynamics), which defines the PDE penalization function. This, as Figure 5 shows, allows us to embed the degradation process to the evolution of the RUL values along with the latent space representation. In both subdatasets, by considering the transition from high RUL values into lower ones as a temporal evolution of a health index, we can use the latent space to determine the health state of the system if it were to be separated by an RUL threshold.

Indeed, using the training dataset, we define a "start of degradation" threshold to separate the health state of the system as either "healthy" or "degraded." Having two different classes allows us to train machine learning classification models based on the results obtained for the latent space representation. This threshold (TH) value needs to be optimized to provide an accurate classifier and ensure that the degradation detection occurs with enough anticipation of the failure. For instance, a TH of 20 cycles considers all points with RUL $\leq$ 20 cycles as degraded, while for RUL $>$ 20 cycles, the system is considered as healthy. However, 20 cycles before failure would not be reasonable since it is too close to the failure event. A TH value of 120, on

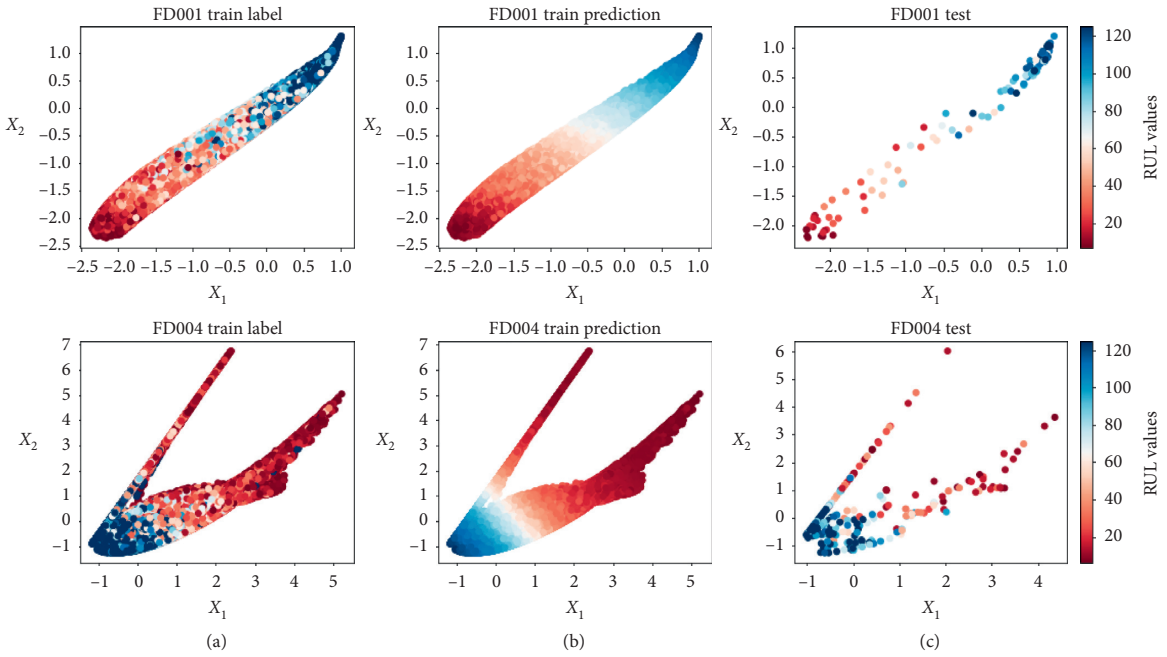


Figure 5: Trained latent variable mapping of RUL training labels (a), RUL training data predictions (b), and RUL testing labels (c).

the other hand, would not be informative since all the RUL labels from the C-MAPSS dataset are commonly defined as lower or equal to 125 cycles. As such, a sensibility analysis to select the TH value is needed for different classifiers. Here, we will focus on six of the most common classifiers in ML approaches. The classifier and TH selection would depend on the accuracy of the model on both the training and validation sets, as well as on the specific system under study.

Figure 6 shows the results for the TH and ML classifier sensibility analysis. All classifiers are trained using the default parameters provided in the sci-kit learn package for Python [45]. The analysis is performed over both subdatasets individually, and results are reported for the training and validation sets. It can be observed that the classification performance decreases with higher TH values for the training and validation sets.

As discussed above, setting a small TH value for the classifier can be impractical from a PHM perspective. Hence, we select 50-cycle TH value for being the longest horizon where classifiers present a 90% accuracy performance while still accounting for 40% of the RUL label range. Tables 2 and 3 detail the performance metrics for all classifiers at TH = 50. Here, the best training performance is obtained with a Nearest Neighbors classifier, which has the lowest validation accuracy and overfits the training set. All remaining classifiers present a similar performance on the training and validation sets with no overfitting. This behavior was also observed in Figure 6. Random forest (RF) stands out among these classifiers due to its low training time and false positive metric. RF is also known for providing good visualization representation that allows us to separate classes visually. As such, we select RF as the classifier for the health state estimator through the latent space representation.

Table 4 presents the results after training ten RF classifiers for each subdataset. Results show a high accuracy for both subdatasets, with all false negatives and false positives below 10%. The FD001 set presents a slight underfitting of its results, which can be associated with the great number of training points generated to include the time dimension during the training process, as discussed in Section 4.

Figure 7 illustrates the trained RF classifier results for both the training and test sets mapped to the latent space representation. Observe the classifier clearly separates a healthy zone (blue) and a degraded zone (red). This classifier is fairly conservative, especially for the FD004 subdataset, mainly due to the selected threshold. It is important to note that, for both subdatasets, the mapping of the test set is consistent with the trained classifier and the RF classifier provides a transition zone (white) which works as a separation boundary between the two defined health states. The trained classifier is not limited to only two classes, and more health states could be added if they are available or needed. These results show that the latent space representation can indeed be used as a health indicator and, as such, can work as a decision variable when planning maintenance policies.

As shown in Figure 5, the shape of the latent variable representation changes from one subset to another. It is then logical that if the proposed framework were to be tested for another system, the visualization and RUL mapping would also be different from the case study discussed. As such, training the selected classifier and setting the corresponding threshold value would vary from system to system. Also, depending on the PHM framework and implementation, setting an optimal threshold value may also depend on other metrics rather than just the models' efficiency. For instance, an optimized TH with a classifier accuracy of 90% might be worth more than having a small TH value (e.g., 5 cycles

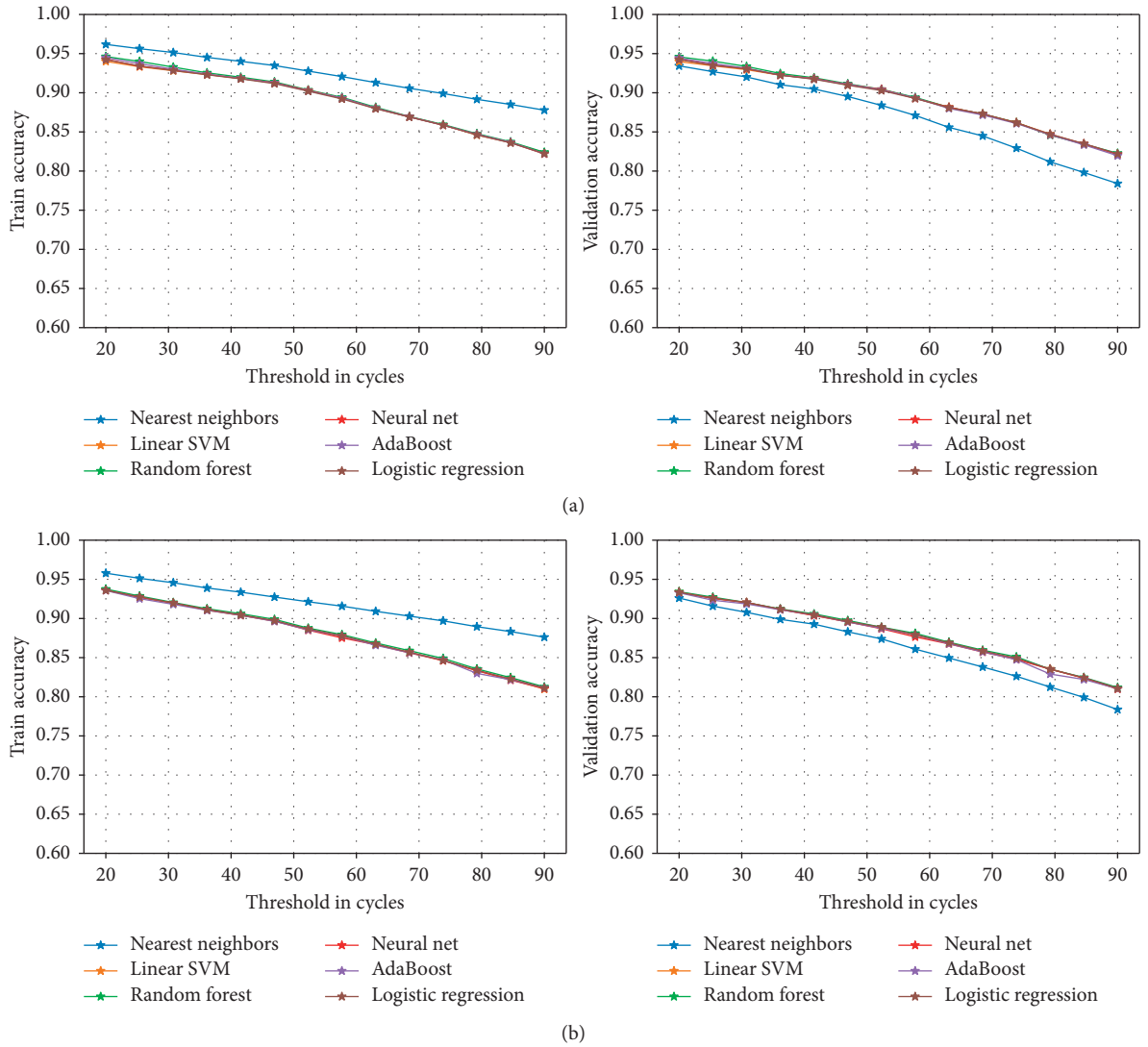


Figure 6: Threshold sensibility analysis for six machine learning classifiers for the health state of the system based on the trained latent variable x. (a) FD001. (b) FD004.

Table 2: Machine learning classifiers performance with TH = 50 for the FD001 subdataset.

ML classifier	Set	Accuracy	False negative	False positive	$F1$-score	Recall	Training time
Nearest neighbors	Train	93.0	4.6	2.4	86.3	82.7	3.57
	Validation	88.7	6.7	4.6	78.1	75.0	
Linear SVM	Train	90.6	6.2	3.2	81.3	76.8	69.37
	Validation	90.7	6.1	3.2	81.6	77.2	
Random forest	Train	90.7	6.7	2.6	81.1	74.9	0.81
	Validation	90.6	6.7	2.7	81.0	75.1	
Neural network	Train	90.6	6.1	3.3	81.3	77.1	10.43
	Validation	90.7	6.0	3.3	81.7	77.5	
AdaBoost	Train	90.6	6.5	3.0	81.0	75.7	3.80
	Validation	90.5	6.5	3.0	81.1	75.9	
Logistic regression	Train	90.5	6.0	3.5	81.3	77.6	1.44
	Validation	90.6	5.9	3.5	81.6	77.9	

Table 3: Machine learning classifiers performance with TH = 50 for the FD004 subdataset.

ML classifier	Set	Accuracy	False negative	False positive	$F1$-score (%)	Recall (%)	Training time (s)
Nearest neighbors	Train	92.4	4.9	2.7	81.9	77.8	4.02
	Validation	87.8	7.2	5.0	71.2	67.7	
Linear SVM	Train	89.0	8.3	2.7	71.3	62.3	94.96
	Validation	89.1	8.2	2.6	72.1	63.0	
Random forest	Train	89.1	7.9	3.0	72.1	64.0	0.98
	Validation	89.2	7.9	2.8	72.8	64.5	
Neural network	Train	89.0	7.5	3.5	72.6	66.0	9.31
	Validation	89.2	7.5	3.3	73.3	66.5	
AdaBoost	Train	88.8	7.8	3.5	71.7	64.7	4.82
	Validation	88.9	7.8	3.3	72.4	65.1	
Logistic regression	Train	89.0	8.0	2.9	71.7	63.4	1.53
	Validation	89.2	8.0	2.8	72.6	64.1	

Table 4: Classification metrics for random forest models with a 95% confidence interval.

		Accuracy (%)	False negative (%)	False positive (%)	$F1$ score (%)	Recall score (%)
FD001	Train	90.7 ± 0.0	6.6 ± 0.2	2.6 ± 0.2	81.1 ± 0.1	74.9 ± 0.8
	Test	92.9 ± 91.2	3.1 ± 0.6	4.0 ± 0.6	89.4 ± 1.8	90.6 ± 1.8
FD004	Train	89.1 ± 0.2	7.8 ± 0.2	2.9 ± 0.2	72.3 ± 0.3	64.3 ± 0.9
	Test	86.7 ± 0.3	9.1 ± 0.3	4.1 ± 0.4	77.5 ± 0.5	71.5 ± 0.9

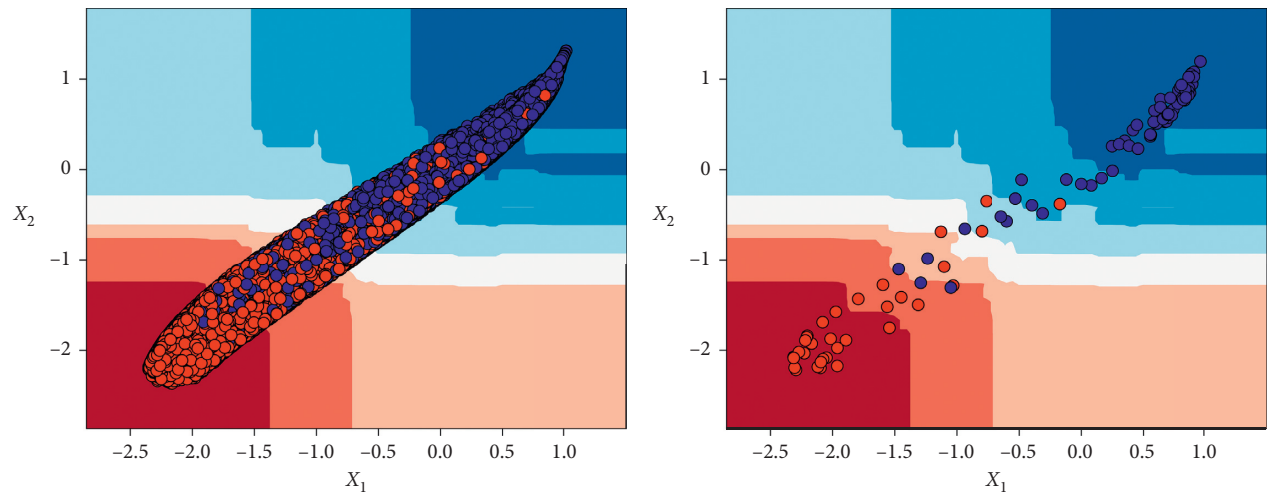


Figure 7: Continued.

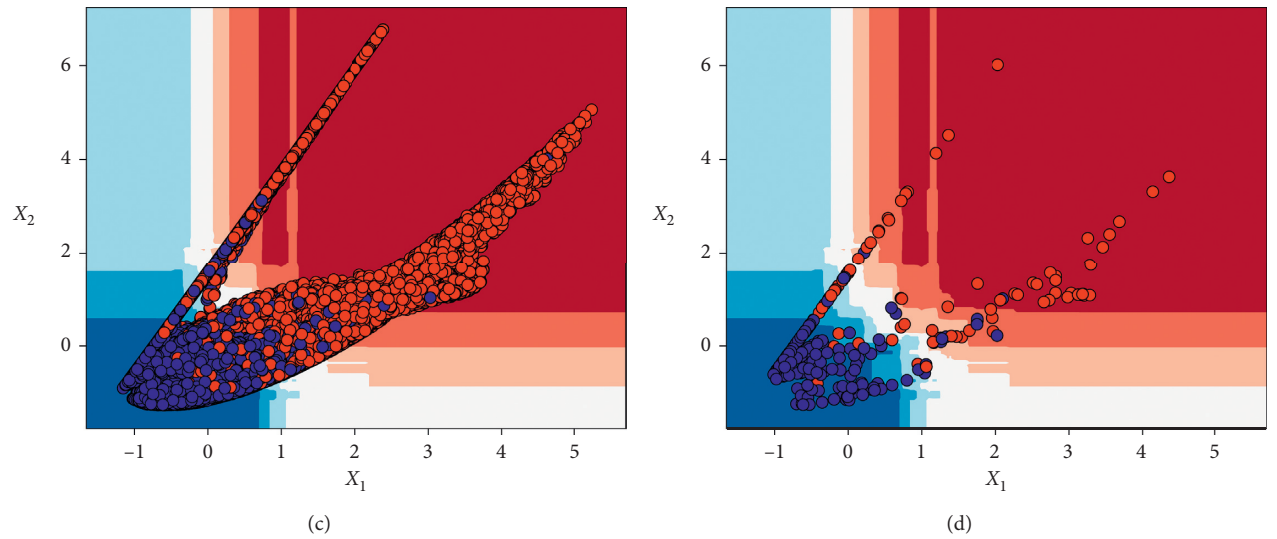


Figure 7: Latent variable classifier decision zones based on a trained random forest model for the FD001 and FD004 subdatasets: degraded state (red) and healthy state (blue). (a) FD001 train. (b) FD001 test. (c) FD004 train. (d) FD004test.

before failure) with a 99% classifier accuracy. An online implementation of the model, along with the classifier, would allow a real-time evaluation of the system's operational conditions. This classifier can be further complemented with the remaining stages of the framework presented in Figure 1, that is, the PDE dynamics N and the RUL estimation. These additional outputs provide information on the system and can be used to create new metrics, rather than just base the results on an RUL value. Thus, this framework creates the opportunity to make better-informed decisions for the maintenance scheduling of complex systems.

Recent DL research works using the C-MAPSS dataset as a case study have focused on feature extraction to improve models' performance, supervised health state estimation, and optimal RMSE values. For instance, Berghout [46] used a denoising autoencoder as a feature extractor coupled with an update selection strategy to determine the training sequences used in an extreme learning machine (ELM) prognosis model. Here, only the FD001 subdataset was trained. Due to the feature extraction process and the ELM prognosis model, this framework contains a high number of trainable parameters and its good performance is likely to be case specific. This model does not provide classification nor visual interpretation. Another example is presented by [30] where a multitask deep CNN is proposed for simultaneous health state and RUL estimation. This model requires manually creating health state labels, which introduces bias to the model, and it does not provide any interpretation of the model. The dual estimation also increases the number of trainable parameters significantly. Results for these configurations and other traditional DL applications to PHM are compared in Table 5. Up to date, there are no frameworks that can provide prediction, classification, and visual representation at the same time. The lower performance on RUL estimation could then be viewed as a tradeoff between prognosis and interpretability of the model. Furthermore, the fewer parameters of our proposed model avoid overfitting problems and, as it was discussed, adapting the framework to other case studies is straightforward. Here, it is important to remark that the proposed framework can be adapted to consider a PBM, when available.

Table 5: Comparison with the state-of-the-art results for RUL RMSE values and state of health classification accuracy for the FD001 and FD004 test sets.

	FD001		FD004	
Model	RMSE	Accuracy	RMSE	Accuracy
ANN	19.62	—	24.35	—
RNN	13.36	—	24.06	—
DOS-ELM [46]	12.29	—	—	—
MT-CNN [30]	12.48	0.71	19.98	0.84
PDE-PHM	17.14	0.92	25.58	0.86

6. Conclusions

This paper presented a framework with the first application of PINN applied to PHM in complex systems. The proposed framework allows the interpretation of the degradation dynamics through a latent space representation and, thus, it is a promising alternative for physics-informed model applications for complex systems. The framework comprises deep neural networks with a total of 1,066 parameters, which is considerably smaller than more complex architectures by at least two orders of magnitude. This contributes to low training and evaluation times while preventing overfitting and makes it a suitable approach to be deployed both online and on mobile devices. This framework establishes a relationship between the time and sensor variables with the degradation of a PDE-like penalization function. We have shown that the obtained two-dimensional latent space acts

as a health indicator of the degradation process of the system, which also can be visually interpreted for engineering purposes as well as a health state classifier through an ML model. Additionally, the proposed framework addresses two major challenges in DL techniques applied to PHM, namely, the use of time as an input variable and the interpretation of the operational conditions from an engineering point of view. This paper takes a step towards bridging the gap between statistic-based PHM and physics-based PHM by providing models that do not need ad hoc and third-party software to interpret its results and it is directly linked to the degradation process of the system. The presented framework is flexible because it can integrate available degradation processes into the training process if these are available. The framework opens many doors to applying these algorithms to real complex systems, especially on maintenance and preventive assessments.

Data Availability

For this study, the C-MAPSS dataset was used. The dataset is publicly available at https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was carried out as a part of the Pipeline System Integrity Management Project, which was supported by the Petroleum Institute, Khalifa University of Science and Technology, Abu Dhabi, UAE. Sergio Cofre-Martel would like to thank the Agencia Nacional de Investigacion y Desarrollo (ANID/Conicyt) for its support through the Becas de Doctorado en el Extranjero, Becas Chile Program.

References

[1] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: a systematic review from data acquisition to RUL prediction," *Mechanical Systems and Signal Processing*, vol. 104, pp. 799–834, 2018.

[2] B. Rezaeianjouybari and Y. Shang, "Deep learning for prognostics and health management: state of the art, challenges, and opportunities," *Measurement*, vol. 163, Article ID 107929, 2020.

[3] A. Cubillo, S. Perinpanayagam, and M. Esperon-Miguez, "A review of physics-based models in prognostics: application to gears and bearings of rotating machinery," *Advances in Mechanical Engineering*, vol. 8, no. 8, pp. 1–21, 2016.

[4] Y. Li, K. Liu, A. M. Foley et al., "Data-driven health estimation and lifetime prediction of lithium-ion batteries: a review," *Renewable and Sustainable Energy Reviews*.vol. 113, 2019.

[5] L. Liao and F. Köttig, "Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction," *IEEE Transactions on Reliability*, vol. 63, no. 1, pp. 191–207, 2014.

[6] A. Al-Dulaimi, S. Zabihi, A. Asif, and A. Mohammadi, "A multimodal and hybrid deep neural network model for remaining useful life estimation," *Computers in Industry*, vol. 108, pp. 186–196, 2019.

[7] L. Liao, W. Jin, and R. Pavel, "Enhanced restricted Boltzmann machine with prognosability regularization for prognostics and health assessment," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7076–7083, 2016.

[8] X. Li, W. Zhang, and Q. Ding, "Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction," *Reliability Engineering System Safety*, vol. 182, pp. 208–218, 2019.

[9] O. Fink, Q. Wang, M. Svensén, P. Dersin, W.-J. Lee, and M. Ducoffe, "Potential, challenges and future directions for deep learning in prognostics and health management applications," *Engineering Applications of Artificial Intelligence*, vol. 92, Article ID 103678, 2020.

[10] J. Ma, S. Xu, Y. Ding et al., "Cycle life test optimization for different Li-ion power battery formulations using a hybrid remaining-useful-life prediction method," *Applied Energy*, vol. 262, Article ID 114490, 2020.

[11] Y. Zhou, M. Huang, and M. Pecht, "Remaining useful life estimation of lithium-ion cells based on k-nearest neighbor regression with differential evolution optimization," *Journal of Cleaner Production*, vol. 249, Article ID 119409, 2020.

[12] G. Ma, Y. Zhang, C. Cheng, B. Zhou, P. Hu, and Y. Yuan, "Remaining useful life prediction of lithium-ion batteries based on false nearest neighbors and a hybrid neural network," *Applied Energy*, vol. 253, Article ID 113626, 2019.

[13] X. Qiu, W. Wu, and S. Wang, "Remaining useful life prediction of lithium-ion battery based on improved cuckoo search particle filter and a novel state of charge estimation method," *Journal of Power Sources*, vol. 450, Article ID 227700, 2020.

[14] Y. Chen, G. Peng, Z. Zhu, and S. Li, "A novel deep learning method based on attention mechanism for bearing remaining useful life prediction," *Applied Soft Computing*, vol. 86, Article ID 105919, 2020.

[15] L. Xu, P. Pennacchi, and S. Chatterton, "A new method for the estimation of bearing health state and remaining useful life based on the moving average cross-correlation of power spectral density," *Mechanical Systems and Signal Processing*, vol. 139, Article ID 106617, 2020.

[16] J. Zhu, N. Chen, and C. Shen, "A new data-driven transferable remaining useful life prediction approach for bearing under different working conditions," *Mechanical Systems and Signal Processing*, vol. 139, Article ID 106602, 2020.

[17] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp. 167–179, 2018.

[18] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *Proceedings of the 2017 IEEE International Conference on Prognostics and Health Management*, Dallas, TX, USA, June 2017.

[19] Z. Zhao, B. Bin Liang, X. Wang, and W. Lu, "Remaining useful life prediction of aircraft engine based on degradation pattern learning," *Reliability Engineering & System Safety*, vol. 164, no. 457, pp. 74–83, 2017.

[20] X. Li, Q. Ding, and J. Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering and System Safety*, vol. 172, pp. 1–11, 2018.

[21] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?' Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference*

on Knowledge Discovery and Data Mining, New York, NY, USA, August 2016.

[22] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the Advances in Neural Information Processing System*, Long Beach, CA, USA, December 2017.

[23] S. Bach, A. Binder, G. Montavon, F. Klauschen, K. R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS One*, vol. 10, no. 7, Article ID e0, 2015.

[24] H. Miao, B. Li, C. Sun, and J. Liu, "Joint learning of degradation assessment and RUL prediction for aeroengines via dual-task deep LSTM networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5023–5032, 2019.

[25] Y. Yu, C. Hu, X. Si, J. Zheng, and J. Zhang, "Averaged Bi-LSTM networks for RUL prognostics with non-life-cycle labeled dataset," *Neurocomputing*, vol. 402, pp. 134–147, 2020.

[26] A. Elsheikh, S. Yacout, and M. S. Ouali, "Bidirectional handshaking LSTM for remaining useful life prediction," *Neurocomputing*, vol. 323, pp. 148–156, 2019.

[27] L. Bellani, M. Compare, P. Baraldi, and E. Zio, *Towards Developing a Novel Framework for Practical PHM: A Sequential Decision Towards Developing a Novel Framework for Practical PHM: a Sequential Decision Problem solved by Reinforcement Learning and Artificial Neural Networks*, 2020.

[28] Z. Mahmoodzadeh, K. Wu, E. L. Droguett, and A. Mosleh, "Condition-based maintenance with reinforcement learning for dry gas pipeline subject to internal corrosion," *Sensors*, vol. 20, no. 19, pp. 1–31, 2020.

[29] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[30] T. S. Kim and S. Y. Sohn, "Multitask learning for health condition identification and remaining useful life prediction: deep convolutional neural network approach," *Journal of Intelligent Manufacturing*, pp. 1–11, 2020.

[31] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of the COMPSTAT'2010*, Paris, France, August 2010.

[32] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Proceedings of the International 1989 Joint Conference on Neural Networks*, Washington, DC, USA, June 1989.

[33] A. Otto, K. Griewank, and A. Griewank, "On automatic differentiation automatic/algorithmic differentiation view project abs-linear learning by gradient based methods or mixed binary linear optimization view project on automatic diierentiation 1 by on automatic diierentiation," 1997, https://arxiv.org/abs/1502.05767.

[34] M. W. M. G. Dissanayake and N. Phan-Thien, "Neural-network-based approximations for solving partial differential equations," *Communications in Numerical Methods in Engineering*, vol. 10, no. 3, pp. 195–201, 1994.

[35] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 987–1000, 1998.

[36] J. Chen, H. Jing, Y. Chang, and Q. Liu, "Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process," *Reliability Engineering & System Safety*, vol. 185, pp. 372–382, 2019.

[37] A. Listou Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang, "Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture," *Reliablility Engineering and System Safety*.vol. 183, pp. 240–251, 2019.

[38] M. Hou, D. Pi, and B. Li, "Similarity-based deep learning approach for remaining useful life prediction," *Measurement*, vol. 159, Article ID 107788, 2020.

[39] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors (Switzerland)*, vol. 17, no. 2, 2017.

[40] A. Saxena, M. Ieee, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proceedings of the 2008 International Conference on Prognostics and Health Management*, pp. 1–9, IEEE, Denver, CO, USA, October 2008.

[41] A. C. Ian Goodfellow, Y. Bengio, and A. Courville, *Deep Learning Book*, MIT Press, Cambridge MA USA, 2015.

[42] A. Tato and R. Nkambou, "Improving adam optimizer," in *Proceedings of the ICLR 2018 Workshop Submission*, Vancouver, Canada, 2018.

[43] H. Li, W. Zhao, Y. Zhang, and E. Zio, "Remaining useful life prediction using multi-scale deep convolutional neural network," *Applied Soft Computing*, vol. 89, Article ID 106113, 2020.

[44] T. Xia, Y. Song, Y. Zheng, E. Pan, and L. Xi, "An ensemble framework based on convolutional bi-directional LSTM with multiple time windows for remaining useful life estimation," *Computers in Industry*, vol. 115, Article ID 103182, 2020.

[45] F. Pedregosa, A. Gramfort, V. Michel et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning and Research*, vol. 12, 2011.

[46] T. Berghout, L.-H. Mouss, O. Kadri, L. Saïdi, and M. Benbouzid, "Aircraft engines remaining useful life prediction with an adaptive denoising online sequential extreme learning machine," *Engineering Applications of Artificial Intelligence*, vol. 96, Article ID 103936, 2020.

Hindawi
Shock and Vibration
Volume 2024, Article ID 9874938, 1 page
https://doi.org/10.1155/2024/9874938

Retraction

Retracted: Research on Optimizing Selection and Optimizing Matching Technologies of Aeroengine Fan Rotor Blades

Shock and Vibration

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] L. Li, K. Chen, J. Gao, Z. Gao, and J. Liu, "Research on Optimizing Selection and Optimizing Matching Technologies of Aeroengine Fan Rotor Blades," *Shock and Vibration*, vol. 2021, Article ID 5595535, 17 pages, 2021.

Hindawi
Shock and Vibration
Volume 2021, Article ID 5595535, 17 pages
https://doi.org/10.1155/2021/5595535

Research Article

Research on Optimizing Selection and Optimizing Matching Technologies of Aeroengine Fan Rotor Blades

Lili Li, Kun Chen, Jianmin Gao, Zhiyong Gao, and Junkong Liu

State Key Laboratory of Mechanical Manufacturing Systems Engineering, Xi'an Jiao Tong University, Xi'an 710049, China

Correspondence should be addressed to Kun Chen; chenkun@mail.xjtu.edu.cn

Received 2 March 2021; Revised 17 March 2021; Accepted 7 April 2021; Published 29 April 2021

Academic Editor: Tangbin Xia

Aiming at the problem of low resources utilization of rotating blades in the selection process of aeroengine fan rotor blades, this paper takes the first-order bending dispersion, first-order torque dispersion, and gravitational moment difference of rotor blades as the selection criteria and takes the minimum remaining blades as the optimization goal. An intelligent selection algorithm of blades based on the collocation degree of blades is proposed and achieves the efficient selection and full utilization of rotating blades. Aiming at the problem of multiple installations and multiple adjustments and low assembly success rate of fan rotor blades, this paper takes the gravity moment difference of the two blades at the diagonal position of 180° as the constraint and takes the minimum residual unbalance as the optimization objective, adopts the improved simulated annealing algorithm to optimize the assembly sequence of rotating blades, and greatly reduces the residual unbalance of blades, which is beneficial to reduce the number of assembly adjustments of blades. The optimizing selection and optimizing matching methods of rotating blades realize the full utilization and efficient assembly of blades and lays a foundation for the reliability and robustness of the assembly quality and service performance of blades.

1. Introduction

In order to improve the balance quality of the aeroengine fan rotor, rotating blades need to be selected and matched according to the natural frequency and the gravitational moment before installation. The selection matching work before assembly of rotor blades mainly includes two parts: ① select rotating blades from the blade database according to the natural frequency dispersion and the gravity moment difference. The goal of selecting blades is to realize the efficient selection and full utilization of rotor blades; ② when the selection of the blade is completed, the assembly sequence of rotating blades must be planned with the goal of the smallest remaining unbalance according to the gravitational moment of rotating blades. The selection-matching work of rotor blades of the aeroengine fan directly determines the balance quality of the product and affects the service performance of the product. Efficient rotating blades selection-matching technology can improve the resources utilization rate of rotating blades and the reliability of the balance quality, increase the success rate of rotating blades assembly, and reduce the number of installation and adjustment of rotating blades.

The constraint of rotor blade optimization-selection is not to exceed the given natural frequency dispersion and gravitational moment difference. The optimization goal is to minimize the number of remaining blades in the candidate library. Therefore, the optimization-selection process of rotating blades is actually an optimization process with constraints. There are not many researches on the optimization-selection of rotating blades, but the problem of rotor blade optimization-selection is essentially an engineering optimization problem. Practical engineering optimization problems often have many characteristics such as complexity, nonlinearity, constraints, and difficulty in modeling. Traditional optimization methods (such as simplex method, Newton method, etc.) need to traverse the entire search space and cannot complete the search in a short time, and it is easy to produce "combination explosion" of searching [1]. Therefore, seeking efficient optimization methods has

become one of the main research contents to solve engineering optimization problems. A lot of progress has been made in the research of optimization methods for optimization problems. It mainly includes genetic algorithm that imitates the biological evolution mechanism of nature [2], differential evolution algorithm that optimizes the search through cooperation and competition between individuals in a group [3], immunity algorithm that simulates the learning and cognitive function of biological immunity system [4], ant colony algorithm that simulates the collective path-finding behavior of ants [5], particle swarm algorithm that simulates the swarm behavior of birds and fish groups [6], simulated annealing algorithm [7] derived from the annealing process of solid matter, tabu search algorithm that simulates the memory process of human intelligence [8], neural network algorithm that simulates the behavioral characteristics of animal neural network [9], etc. [10]. These algorithms are developed by simulating or revealing certain natural phenomena, processes, or intelligent behaviors of biological groups. These optimization algorithms have the advantages of being simple, versatile, and convenient for parallel processing [11] and provide reference for the solution of the optimization-selection of rotating blades.

The assembly sequence planning of rotating blades is a nonnegligible part of the selection-matching work of rotating blades and directly determines the static balance quality of the rotor. However, the assembly sequence planning problem has NP-hard characteristics. In order to search for all feasible assembly sequence schemes and find the optimal assembly sequence, the complexity of searching for the optimal sequence will increase toward the direction of exhaustive search, and it is difficult to obtain a relatively optimal assembly sequence in a short time; this challenge has become one of the important driving forces to encourage the research of computerized assembly sequence planning [12]. In order to solve the ASP (Assembly Sequence Planning) problem, researchers used a variety of optimization algorithms to optimize the ASP problem, such as Ant colony optimization algorithm (ACO) [13], genetic algorithm (GA) [14, 15], immune algorithm (IA) [16], neural networks (NN) [17], scatter search algorithm (SSA) [18], and other heuristic methods [19–21]. At present, researchers have made remarkable progress in solving ASP optimization problems, but there are still some problems that need to be solved urgently. One of the most important problems is that it is difficult to obtain a relatively optimal assembly sequence in a short time. This problem has prompted researchers to introduce or improve various algorithms to improve the solving accuracy, robustness, and efficiency of the ASP problem.

The above research results provide a reference for the optimization-selection and optimization-matching of rotor blades. The rotor blade is the core component of the aeroengine fan rotor, and its balance quality is the main criterion for the assembly quality of rotating blades [22]. The "selection" and "matching" of rotating blades before assembly directly determine the "installation" and "adjustment" during the assembly process of rotor blades, as well as the balance quality and service performance of the rotor after the assembly is completed. Therefore, in order to achieve the optimizing selection and optimizing matching of aeroengine fan rotor blades, this paper proposes an intelligent selection algorithm based on the collocation degree of blades to realize efficient selection and full utilization of rotor blades. The improved simulated annealing algorithm is used to optimize the assembly sequence of rotor blades, so that the residual unbalance of rotor blades can be as small as possible, and the success rate of one-time assembly of the rotor blades can be improved, and the number of installation and adjustment of rotor blades can be reduced. Finally, ensure that the goals of optimizing selection and optimizing matching of rotor blades before assembly, optimized assembly sequence and less adjustment during assembly, and reliable and stable balance quality after assembly are achieved.

2. Analysis of the Problem of Selection-Matching of Rotor Blades of the Aeroengine Fan

In order to ensure the robustness and reliability of the balance quality and service performance of the aeroengine fan rotor, the company currently guarantees it mainly from the following two aspects:

(1) "Selection" and "matching" before assembly of aeroengine fan rotor blades. Aeroengines are mass-produced, and the blade database often contains hundreds of thousands of blades. If the blades are not selected, and the blades required by the fan rotor will be taken out randomly from the blade database, the first-order bending dispersion, first-order torque dispersion, and gravitational moment difference of the blades will lose control, which will not only be difficult to guarantee the remaining unbalance of rotating blades but also increase the difficulty of dynamic balancing of rotor blades. Therefore, before assembling rotating blades, blades must be selected according to certain selection criteria. The more the number of blades selected from the blade database, the remaining blades in the blade database will be fewer, and the utilization rate of blade resources will be higher. The blade selection is based on the dispersion of the first-order bending and the first-order torque, and the gravitational moment difference of the largest blade and the smallest blade of the rotor. The purpose is to ensure that the selected blades are as uniform as possible, and the characteristic gap between each other cannot be too big, laying the foundation for subsequent blade assembly sequence planning and the assurance of balance quality. "Matching" is to plan the assembly sequence of rotating blades and make the remaining unbalance of the rotor as small as possible. However, the remaining unbalance of rotating blades achieved by the current assembly sequence planning technology of the enterprise is generally too large, which causes rotor to be out of tolerance due to the assembly process error.

(2) "Installation" and "adjustment" in the assembly process of aeroengine fan rotors. "Installation" is to complete the actual assembly of rotating blades according to the assembly sequence planned before assembly. "Adjustment" means that in the process of assembly, once the static unbalance of the rotor occurs, the assembly of rotating blades can reach the static balance quality by adjusting installation position and angle of the blades. At present, the success rate of one-time assembly of fan rotor blades in enterprises is not high, and the problem of multiple installations and multiple adjustments is obvious. One of the reasons is that although the residual unbalance obtained by the enterprise is within the design range, it is generally too large, which results in excessive residual unbalance due to assembly process errors. In order to compensate the excessive residual unbalance, the adjustment of rotating blades has to be carried out.

3. Ideas of Optimizing Selection and Optimizing Matching of Rotating Blades

Efficient selection and full utilization of blades are the two most critical goals for blade selection. The company initially relied on manual selection, which was inefficient and relied on workers' experience. Moreover, manual selection of rotating blades could only ensure that 37%–46% of rotating blades were picked out, and more than half of the rotating blades were left in the blade database and were imported into the next batch of new blades. The accumulation of the remaining rotating blades over time has eventually led to more and more blades becoming "nail households" in the blade database, causing idle and waste of blade resources. Later, the company introduced new technologies to select rotor blades, which greatly improved the efficiency of blade selection. However, the current blade selection technology of the enterprise can only achieve a utilization rate of 65%–74% of blade resources, and about 30% of remaining blades still are like "snowballs"; batches of backlogs of rotating blades are accumulated in the blade database, causing idle and waste of blade resources. Therefore, based on the actual engineering needs of the enterprise, this article proposes an intelligent and efficient blade selection algorithm.

After the blade selection is completed, the assembly sequence of selected blades must be planned. Although the remaining unbalance of rotor blades obtained by the company's current assembly sequence planning technology does not exceed the design value, the overall remaining unbalance is generally too large. In actual assembly, the static balance of the rotor is difficult to meet the tolerance because of assembly errors, which leads to the problem of multiple installation and multiple adjustment in the process of assembly of rotor blades. Therefore, this paper adopts the improved simulated annealing algorithm to provide optimized assembly sequence for the rotating blades.

Aiming at the problem of low resource utilization of rotating blades in the selection process of blades, this paper proposes an intelligent optimization algorithm for selecting rotating blades based on the collocation degree. Aiming at the problem that the assembly sequence currently planned by the enterprise is prone to static unbalance because of assembly process errors, resulting in multiple installations and multiple adjustments of rotating blades, this paper adopts the improved simulated annealing algorithm to plan the assembly sequence of rotating blades. The framework of optimizing selection and optimizing matching of rotating blades is shown in Figure 1; b1, b2, . . ., b302 represent blade1, blade2, . . ., blade302.

4. Solution and Analysis of the Optimizing Selection Problem of Aeroengine Fan Rotor Blades

4.1. The Establishment and Solution of the Intelligent Optimization Algorithm for Selecting Rotating Blades Based on the Collocation Degree of Rotating Blades. The aeroengine fan rotor has three-stage blades; the research in this paper takes the selection of the first-stage blades as an example. The first-stage blade database with related data of a total of 302 blades obtained from the enterprise is shown in Table 1. A rotor requires 28 first-stage blades; there are 302 blades in the first-stage blade database in this article, and a maximum of 280 blades are picked out and assembled to form 10 rotors. The rules of selecting blades are shown in Table 2, and the calculation methods of dispersion and gravitational moment difference are shown in formulas (1)–(3). The goal of selecting blades is that the fewer remaining blades, the better, and the more fan rotors that can be assembled with the selected blades, the better. The calculation method of remaining blades is shown in equation (4); N represents the number of rotors assembled by selected blades.

$$\text{the first} - \text{order bending dispersion} = \frac{\max(\text{the first} - \text{order bending}) - \min(\text{the first} - \text{order bending})}{\min(\text{the first} - \text{order bending})} \leq 0.06, \quad (1)$$

$$\text{the first} - \text{order torque dispersion} = \frac{\max(\text{the first} - \text{order torque}) - \min(\text{the first} - \text{order torque})}{\min(\text{the first} - \text{order torque})} \leq 0.08, \quad (2)$$

$$\text{gravitational moment difference} = \max(\text{gravitational moment}) - \min(\text{gravitational moment}) \leq 6000, \quad (3)$$

$$\text{the remaining blades} = 302 - 28 * N. \quad (4)$$

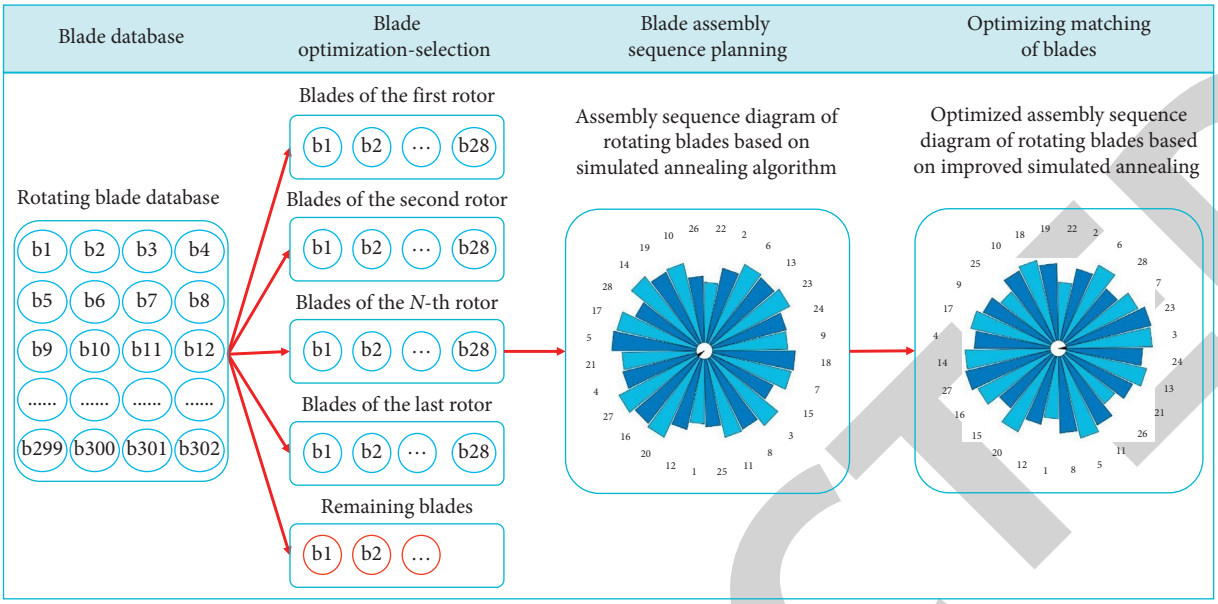


Figure 1: The framework of optimizing selection and optimizing matching of rotating blades.

Table 1: The rotating blade database of the first-stage fan rotor.

No.	First-order bending	First-order torque	Gravitational moment (g*mm)	No.	First-order bending	First-order torque	Gravitational moment (g*mm)
1	127	616	276180	11	122	670	275320
2	128	603	275040	12	133	655	273820
3	130	613	276560	13	127	665	270440
4	129	624	281520	14	137	644	278460
5	136	616	275280	15	125	680	276120
6	129	624	280900	16	136	679	273660
7	124	661	272900	. . ....	. . ....	. . ....	. . ....
8	121	666	275820	300	134	670	276340
9	121	688	278380	301	136	656	276000
10	125	652	272860	302	128	664	277020

Table 2: The selection rules of the first-stage rotor blades.

The number of blades of the first-stage rotor	The first-order bending dispersion	The first-order torque dispersion	Gravitational moment difference
28 blades/group	≤ 0.06	≤ 0.08	6000g*mm

The collocation of rotating blades: if the first-order bending dispersion, the first-order torque dispersion, and the gravitational moment difference of two rotating blades meet in formulas (1)–(3), we call these two blades collocation. The number of collocations of a rotating blade: there are 302 blades in the blade database in this article, and each blade can form 301 pairs of blades with the rest. Take No.i blade as an example, if among the 301 pairs of blades, there are *ni* pairs of blades satisfying formulas (1)–(3), we call the number of collocation of the No.i blade as n_i. By analogy, these 302 blades, each of which has its own number of collocation, are recorded as $n_1, n_2, n_3, \ldots, n_i, \ldots, n_{302}$. Before calculating the collocation degree of each blade, we must first calculate the number of collocation of each blade: $n_1, n_2, n_3, \ldots, n_i, \ldots, n_{302}$. Then, we must calculate the collocation degree of each blade according to the number of collocation of each blade: $P_1 = n_1/\sum_{a=1}^{302} n_a$, $P_2 = n_2/\sum_{a=1}^{302} n_a$, $P_3 = n_3/\sum_{a=1}^{302} n_a, \ldots, P_i = n_1/\sum_{a=1}^{302} n_a, \ldots, P_{302} = n_{302}/\sum_{a=1}^{302} n_a$. The collocation degree of each blade is the probability of each blade being selected, which reflects the ability of a blade matching other blades; the more the number of blades that can be matched with the blade, the higher the collocation degree of the blade. Otherwise, the fewer the number of blades that can be matched with the blade, the lower the collocation degree of the blade. In the blade selection process, the blade with the higher collocation degree has a greater probability of being selected, and the blade with the lower collocation degree may become the final remaining blade.

At present, the company's blade selection technology can only achieve a utilization rate of 65%–74% of blade resources, and there will still be about 30% remaining rotating

blades, which are backlogged in the blade database. These remaining blades will be like "snowballs," creating a backlog, batch by batch, in the blade database, causing idle and waste of blade resources. Based on the actual engineering needs of the enterprise, this paper establishes an intelligent algorithm for selecting blades, takes the fewest remaining blades (which means that the most fan rotors can be assembled by the selected blades) as the optimization goal, and takes the efficiency of blade selection into account. In order to realize the intelligent selection of rotating blades, there are two ideas for programming the optimization algorithm of selecting blades, as shown in Figure 2. Idea one is based on the random selection method to select rotating blades. Each blade in the blade database has the same probability of being selected. For each new blade selected, the constraint conditions must be calculated and verified. If it is satisfied, the new blade will be picked out successfully; otherwise, randomly select a blade from the candidate library again until the constraint conditions are met. The disadvantage of this method is that it is not sure how many times the No. N blade is selected before it can be successfully selected, causing the selection of rotating blades to be random and unstable. The second idea selects rotor blades based on the collocation degree of rotating blades. Each blade has a different probability of being picked out, blades with a higher collocation degree are more likely to be picked out first, and those blades with a lower collocation degree are more likely to become final remaining blades. All the blades in the blade database will be combined into pairs to form many pairs of blades in idea 2, and these blades' collocation degree will be calculated before being selected. If the pair of blades meets the constraint conditions, the pair of blades can be matched and recorded as "1"; otherwise, it is recorded as "0". The collocation matrix formed by these pairs of blades is a symmetric matrix, as shown in Figure 2(c), and the diagonal "1" is not considered because the blade cannot form a pair with itself in actual application. Every time a new blade is selected, there is no need to repeat the calculation in idea 2; it is only necessary to determine whether all the pairs of blades composed by the new blade and the selected blades are matched. If they are all matched, the new blade is successfully picked out; for example, when selecting the No. 4 blade of the rotor, if all the three pairs of blades 4–3, 4–2, 4–1 meet the constraint conditions, which means all the three pairs of blades are matched, then the No. 4 blade is successfully picked out. The advantage of the second idea is that it can complete the calculation of the collocation degree matrix of blades before selecting blades, and when selecting the blades, it is only necessary to judge whether the pairs of blades formed by the new blade and the selected blades are all matched; there is no repeated calculation, idea 2 has a certain selection law, and the probability of each blade being selected is different. Compared with the first idea, the blade selection method based on the second idea is more reliable and robust.

Compile two optimization algorithms of selecting rotating blades according to idea 1 and idea 2, respectively, and run the two algorithms multiple times and compare the solution results (as shown in Figure 3). The optimization algorithm based on the collocation degree of rotating blades can pick out more blades and cost less time than the optimization algorithm of selecting blades based on idea 1; the solution effects of the later optimization algorithm of selecting blades are unstable, especially the solution time, which is very volatile, and the robustness is obviously inferior to the optimization algorithm based on the collocation degree of blades. Therefore, this paper adopts the optimization algorithm based on the collocation degree of rotating blades to select rotating blades.

The flowchart of the optimization algorithm of selecting rotating blades based on the collocation degree of blades is shown in Figure 4. The specific process of the optimization algorithm is shown as follows:

Step (1): combine 302 blades in pairs to form $302 \times 301 \div 2 = 45451$ pairs of blades, and judge whether the first-order bending dispersion, first-order torque dispersion, and gravitational moment difference of these 45451 pairs of blades meet the blade selection rules; if the selection rules are met, the pair of blades is marked as "1"; otherwise, it is recorded as "0". The collocation matrix formed by these pairs of blades is a symmetric matrix, as shown in of Figure 2(c); the diagonal "1" is not considered because the blade cannot form a pair with itself in actual application.

Step (2): establish a candidate library and a finished product library of rotating blades. The candidate library stores 302 blades waiting to be selected, and the finished product library stores blades that meet selection rules and have been picked out. For example, if you can pick out rotating blades needed by 5 rotors from 302 blades, the finished product library will store these rotating blades needed by 5 rotors.

Step (3): among the 302 blades in the blade database, each blade can form 301 pairs of blades with the remaining blades. Take the No. i blade as an example, assume that the No. i blade and the remaining blades are composed of 301 pairs of blades, There are n_i pairs of blades marked as 1, the No. i blade is marked as n_i, which means that it can combine with the other blades into n_i pairs of blades marked as 1, and its number of collocations is n_i. By analogy, count the number of collocations of each rotating blade and record them as $n_1, n_2, n_3, \ldots, n_i, \ldots, n_{302}$, then the probability of the No. i is $P_i = n_i / \sum_{a=1}^{302} n_a$; by analogy, get the probability of each blade, respectively, recorded as $P_1, P_2, P_3, \ldots, P_i, \ldots, P_{302}$. The probability of each blade obtained in this step is the collocation degree of each blade, reflecting the ability of a blade to combine with other blades to form pairs of blades that meets selection rules.

Step (4): use the roulette method to select the first blade of the first rotor according to the collocation degree (probability) of each blade calculated in step (3). Then, continue to use the roulette method to select the second blade of the rotor, check whether the pair of blades formed by the second blade and the first blade is marked as 1 in step (1). If it is marked as 1, the second

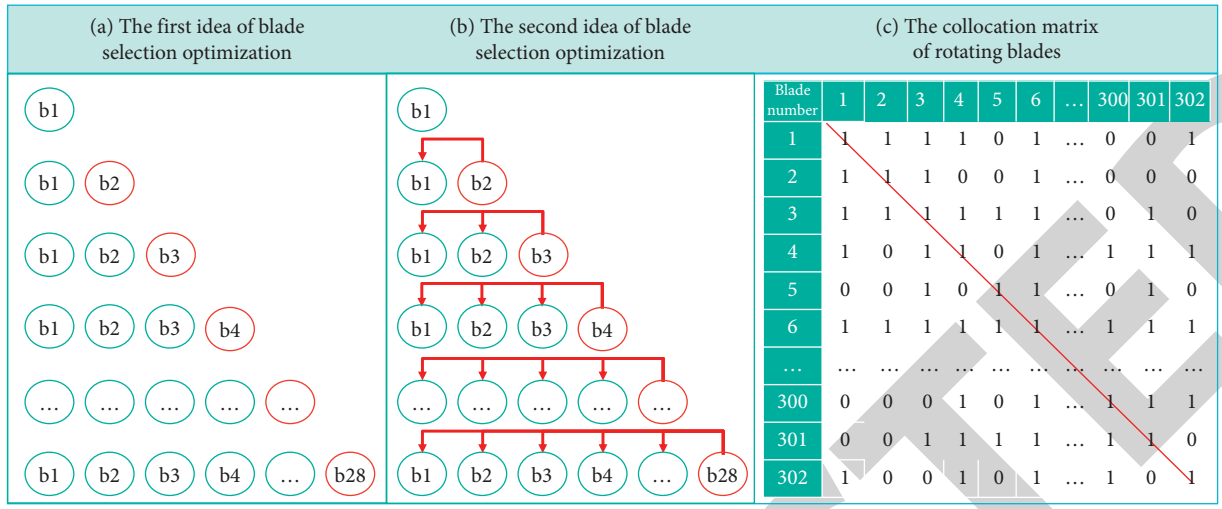


Figure 2: Two ideas of establishing blade optimization-selection algorithm and the collocation matrix of blades.

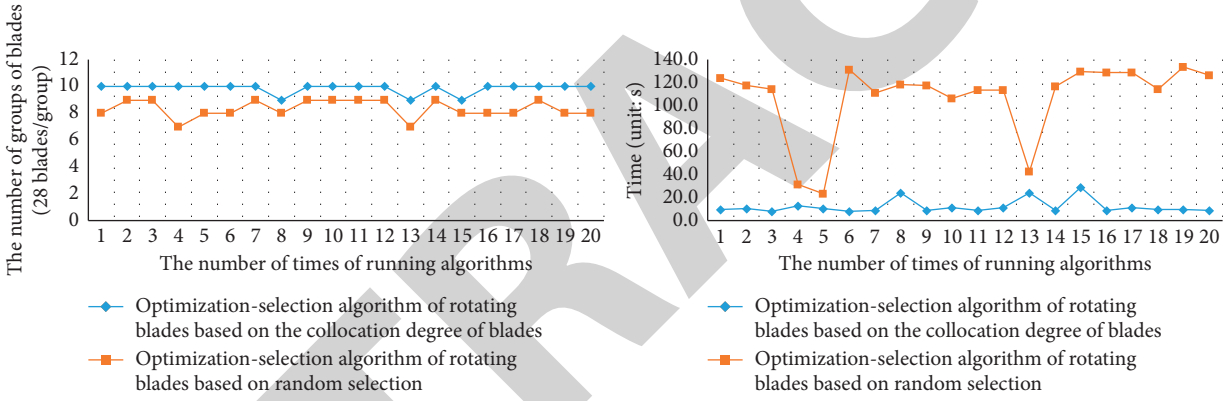


Figure 3: Comparison of the solution effects of the two algorithms based on two ideas.

blade is picked out successfully, and then pick out the third blade in the same way, check whether the two pairs of blades (No. 3 blade and No. 2 blade, No. 3 blade and No. 1 blade) formed by the third blade and the first two selected blades are all marked as 1. If yes, the selection of the third blade is finished; otherwise, use the roulette method, and according to the probability of each blade, reselect the third blade. That is to say, when selecting the No. N blade, check whether the $n-1$ pairs of blades formed by it and the $n-1$ selected blades before are all marked as 1 in step (1). If they are all marked as 1, the No. N blade is picked out successfully; otherwise, reselect the No. N blade, and so on, until all the blades of the first rotor are picked out successfully. Then, continue to select blades of the second rotor. When selecting blades, every time a blade is picked out, the blade will be put into the finished product library in time. When 28 blades of a rotor are picked out from the candidate library, the collocation degree of the remaining blades in the candidate library must be recalculated. Then, use the same method of selecting blades to pick out blades of the next rotor.

Step (5): when the blade selection process reaches a certain degree, the process of blade selection in step (4) will encounter a "bottleneck", which means it is no longer possible to pick out new blades by the method of selecting blades in step (4). At this time, it is assumed that blades of n rotors have been picked out and put into the finished product library, which are, respectively, recorded as $T_1, T_2, \ldots, T_n$. Randomly select a blade from the blades of No. T_n Rotor, and record it as No. M blade, and use the roulette method to select a blade from the candidate library; check whether the blade can replace the No. M blade of No. T_n rotor of the finished product library. If it can, replace the No. M blade with the blade of the candidate library and put it in the finished product library. At the same time, put the No. M blade into the candidate library; otherwise, continue to select the blade that can replace the No. M blade from the candidate library. After completing the replacement of the No. M blade, proceed to step (4) to select blades of the next rotor.

Step (6): when the process of blade selection encounters the "bottleneck" again, and the new blade cannot be

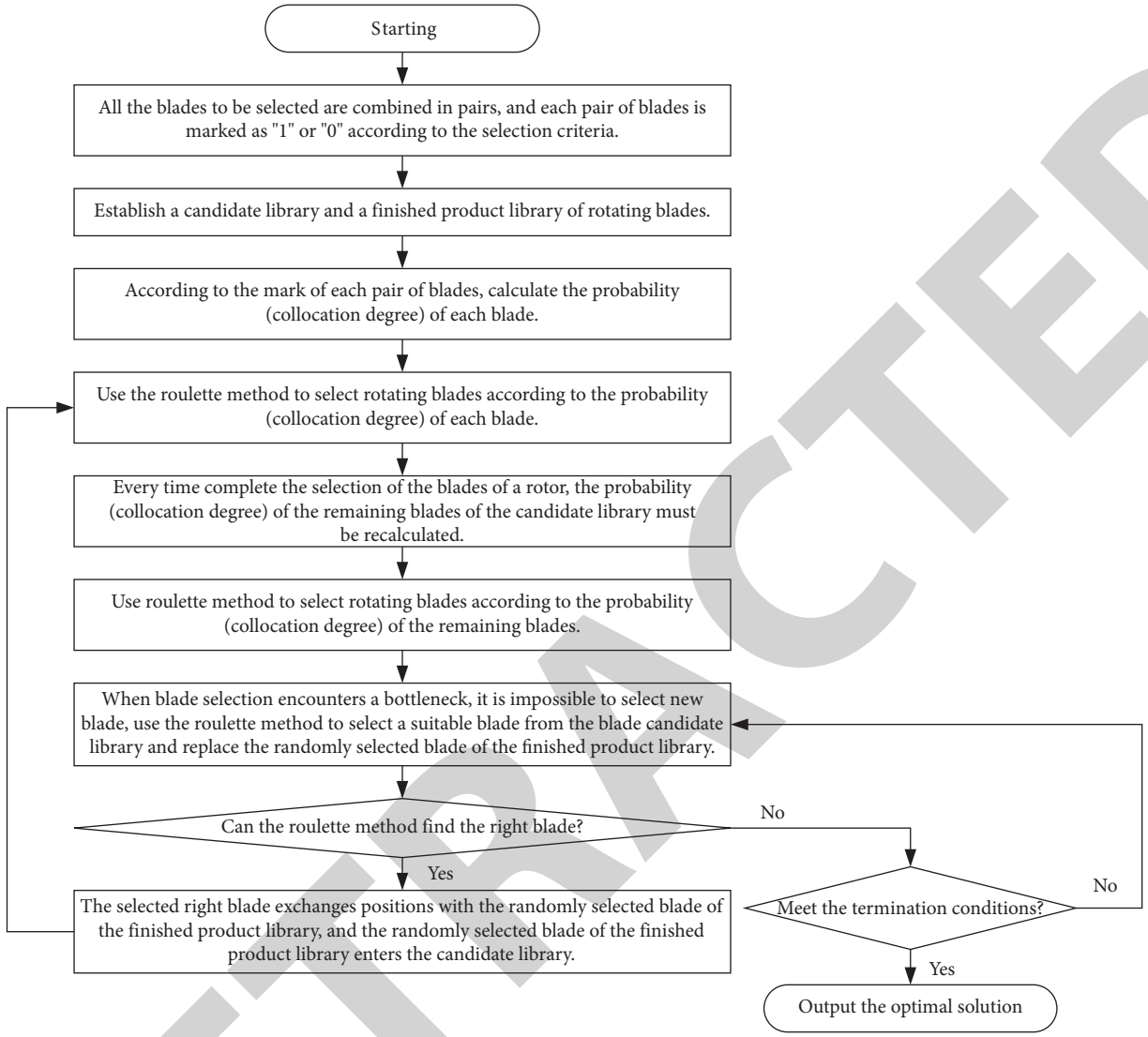


Figure 4: The flowchart of the optimization algorithm of selecting rotating blades based on the collocation degree of blades.

picked out according to the method of selecting blades in step (4), the algorithm goes to step (5), and step (5) is used to solve the problem of the "bottleneck".

Step (7): the number of selected blades and the corresponding blade information are output. The number of rotors that assembled by the selected blades and the corresponding blades of each rotor are output.

The optimization algorithm of selecting rotor blades based on the collocation degree of rotating blades can pick out 10 groups of blades that meet the selection rules; these 10 groups of rotor blades correspond to 10 rotors, and the assembly of these 10 rotors can be completed by these 10 groups of rotor blades. The optimization-selection results are shown in Table 3. There are a total of 302 blades in the rotating blade database of the first-stage rotor; 28 blades are a group of rotor blades corresponding to a rotor, and 302 blades can select 10 groups of rotor blades at most and can assemble 10 rotors. The optimization algorithm picked out 280 blades, and the remaining blades are the fewest, which is 22 blades; blade utilization has reached the theoretical maximum: $280/302 * 100\%$. The distributions of the number of collocations of all the blades before selection and the remaining blades after selection are counted, respectively, as shown in Figure 5; the number of blade collocations refer to the number of pairs that each blade can match with in the remaining 301 blades, as can be known from Figure 5. The number of collocations of most remaining blades is relatively low; only a few remaining blades have a slightly higher number of collocations. Therefore, Figure 5 further explains that blades with a higher collocation degree are more likely to be picked out, while blades with a lower collocation degree tend to become the remaining blades. The higher the number of collocations of the blade, the higher the collocation degree of the blade.

The optimization algorithm of selecting rotor blades based on the collocation degree of rotating blades can pick out 10 groups of blades that meet the selection rules; in order to verify the 10 groups of blades corresponding to 10 rotors, the dispersion and gravitational moment

Table 3: Ten groups of rotating blades (280 blades) selected from the blade database.

No.	Blade no.	First-order bending	First-order torque	Gravitational oment (g*mm)	No.	Blade no.	First-order bending	First-order torque	Gravitational moment (g*mm)
1	222	134	674	281280	...	...	...	...	...
2	50	131	661	278100	254	194	129	662	280920
3	293	135	675	276160	255	287	136	662	279360
4	115	132	669	275700	256	252	135	658	281140
5	96	135	664	281080	257	298	132	695	277460
6	143	134	677	276440	258	57	135	676	278620
7	221	134	675	280220	259	214	129	663	281260
8	203	130	679	278560	260	177	136	678	280620
9	158	134	643	276460	261	167	133	670	278520
10	120	134	667	277300	262	189	130	661	277000
11	178	132	661	278760	263	201	134	675	281200
12	237	135	662	281120	264	140	135	679	278300
13	238	129	676	276840	265	186	129	674	276800
14	174	134	662	278820	266	233	134	676	281080
15	265	129	660	276520	267	225	134	676	280760
16	171	131	667	276080	268	157	130	662	276080
17	213	131	671	281600	269	296	129	644	279540
18	142	136	676	278640	270	147	134	663	277120
19	262	129	662	278140	271	51	131	661	278440
20	44	134	676	277056	272	85	132	661	278040
21	185	134	653	279280	273	289	132	690	277620
22	52	135	661	277100	274	279	130	677	275520
23	285	136	661	276600	275	183	129	663	277460
24	93	130	673	277700	276	235	133	662	275460
25	195	129	673	277360	277	263	136	673	277880
26	116	134	677	275980	278	256	132	645	275680
27	149	130	677	275760	279	133	134	679	276660
28	234	135	659	279840	280	137	131	673	278600

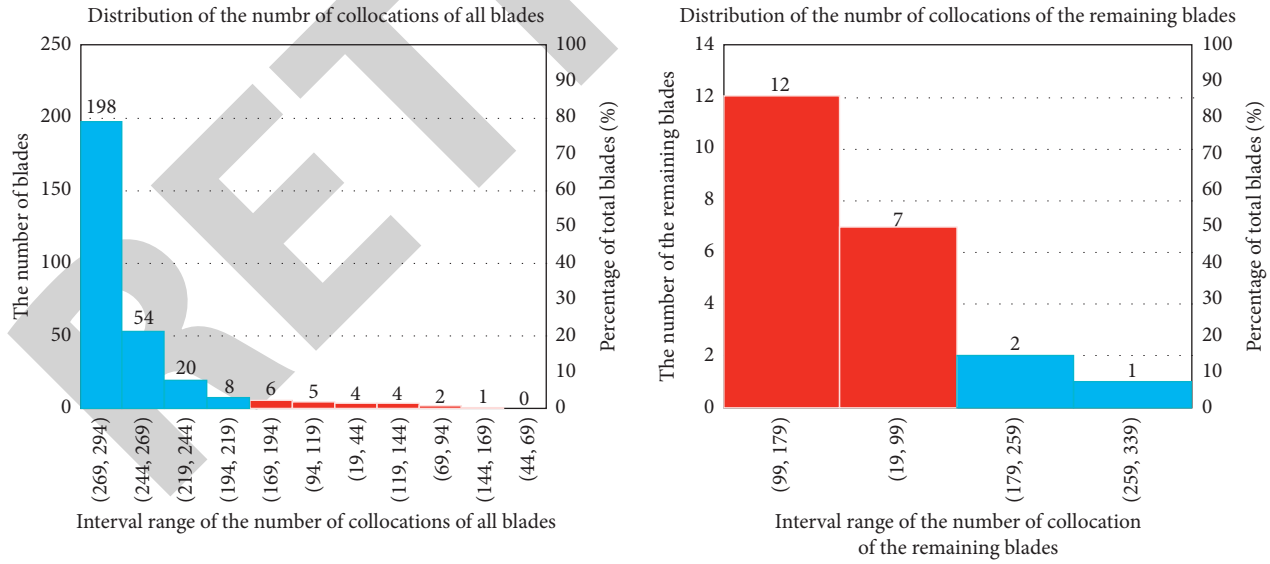


Figure 5: The distributions of the number of collocations of all the blades before selection and the remaining blades after selection.

difference of the selected 10 groups of blades are verified, respectively. The verification results are shown in Table 4. The verification results show that the dispersion and gravitational moment difference of the selected 10 groups of blades are within the specified range, which means that the selected 10 groups of blades by the intelligent optimization algorithm based on the collocation degree of rotating blades meet the requirements of selecting blades. Therefore, the intelligent optimization algorithm has reached the optimal selection goal, and the number of the remaining blades has reached the fewest, and the blade resources are utilized to the greatest extent. The running

Table 4: Ten groups of rotating blades picked out by the algorithm based on the collocation degree of blades.

	First-order bending dispersion	First-order torque dispersion	Gravitational moment difference (g*mm)
Design requirements (blade selection rules)	0.06	0.08	6000
The blade group of the first rotor	0.054	0.056	5900
The blade group of the second rotor	0.054	0.040	6000
The blade group of the third rotor	0.055	0.047	5780
The blade group of the fourth rotor	0.055	0.059	5640
The blade group of the fifth rotor	0.054	0.078	5400
The blade group of the sixth rotor	0.054	0.059	5980
The blade group of the seventh rotor	0.054	0.059	5820
The blade group of the eighth rotor	0.056	0.074	5940
The blade group of the ninth rotor	0.053	0.064	5980
The blade group of the tenth rotor	0.054	0.079	5800

time of the optimization algorithm of selecting blades and the robustness of the results are also two important indicators to measure the quality of the algorithm; therefore, run the optimization algorithm for 20 times, and the running time and optimization results of the optimization algorithm are counted. The statistical results are shown in Table 5. As can be seen form Table 5, the optimization results are very impressive; the minimum running time of the algorithm is 7.7 seconds, the maximum is 28.7 seconds, and the solution efficiency is very high. Therefore, compared with the 65%–74% blade utilization rate currently achieved in the enterprise, the optimization algorithm established in this paper can reach 83%–93% blade utilization rate. Moreover, the solution time of the optimization algorithm based on the collocation degree of blades is very short, which is very convenient for promotion and application in enterprises.

4.2. Comparison and Analysis of the Solution Effect of the Intelligent Optimization Algorithm Based on the Collocation Degree of Rotating Blades and Other Intelligent Optimization Algorithms. The basic idea of the greedy algorithm is to only make the best choice at the moment, the greedy algorithm does not consider the overall optimality, and the choice it makes is only a local optimal choice in a certain sense [23]. The solution of each step of the greedy algorithm is feasible and meets the corresponding constraints; the solution of each step is a local optimal solution and the best solution of all the current feasible solutions. The most obvious disadvantage of the greedy algorithm is that it cannot guarantee that the result obtained is the global optimal solution. However, the greedy algorithm usually does not take up too much time and manpower in the specific solution process. Starting from the purpose of business operation, although the greedy algorithm cannot obtain the global optimal solution, it can find a feasible solution close to the global optimal solution in a short time. Therefore, the solution is close to the global optimal solution sought by the greedy algorithm and can also be accepted by related companies.

The core idea of the simulated annealing algorithm (Simulated Annealing, SA) is derived from the principle of solid annealing. Because of the similarity between the physical annealing process and the combinatorial optimization problem, it was introduced into the field of combinatorial optimization in 1983 [24]. The simulated annealing algorithm selects new solutions according to Metropolis criteria; it not only accepts optimized solutions with excellent performance but also accepts deteriorating solutions with poor performance as a certain probability, prompting the algorithm to jump out of the "trap" of local optimal solutions, thereby ensuring that it can search for the global optimal solution or near optimal solution [25].

In addition to the intelligent optimization algorithm of selecting blades proposed in this paper, this paper also uses the greedy algorithm and the simulated annealing algorithm for selecting blades. The comparison of the solution results and solution efficiency of the three algorithms are shown in Figures 6 and 7 , respectively. The intelligent optimization algorithm of selecting blades based on the collocation degree of blades picks out 9 groups of blades (only 3 times), which can complete the assembly of 9 rotors, but the algorithm picked out 10 groups of blades in the remaining 17 times; therefore, the intelligent optimization algorithm based on the collocation degree of blades has the optimal optimization effect. The solution effect of the greedy algorithm is more stable than that of simulated annealing, but it can only select 9 groups of blades at most, which cannot reach the global optimal solution of 10 groups of blades. This is related to the inherent properties of the greedy algorithm; it cannot find the global optimal solution, but it can obtain a feasible solution close to the global optimal solution in a short time. As shown in Figure 7, the greedy algorithm has the most stable and shortest solution time. It can be seen from Figure 6 that simulated annealing can find the global optimal solution, but the solution result of simulated annealing is the most unstable; it can pick out 10 groups of blades at most, but the smallest is only 1 group of blades being selected. Because the simulated annealing algorithm is based on the Metropolis criterion to select new solutions, it can not only accept the optimized solution with good performance but also accept the deteriorating solution as a certain probability. The advantage of the Metropolis criterion is to promote the algorithm to jump out of the "trap" of the local optimal solution, so as to ensure that it can search for the global optimal solution. But, at the same time, precisely because of accepting the inferior solution as a certain probability during

Table 5: Statistic table of solution time and selection results of rotating blades optimization-selection algorithm based on the collocation degree of blades.

Serial number	Number of groups of blades	Solution time (unit: s)	Serial number	Number of groups of blades	Solution time (unit: s)
1	10	9.6	11	10	8.2
2	10	10.0	12	10	11.1
3	10	7.7	13	9	23.7
4	10	12.3	14	10	8.6
5	10	10.5	15	9	28.7
6	10	7.9	16	10	8.9
7	10	8.5	17	10	11.1
8	9	23.8	18	10	9.0
9	10	8.7	19	10	9.5
10	10	11.2	20	10	8.7

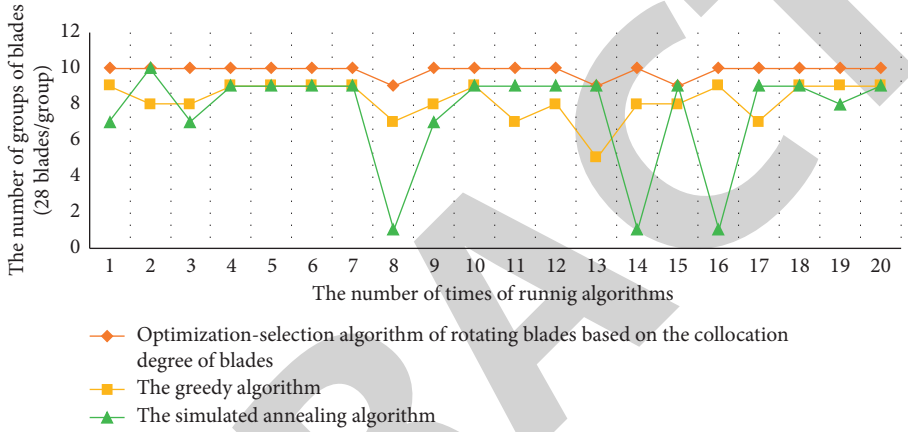


Figure 6: Comparison of solution results of the three algorithms.

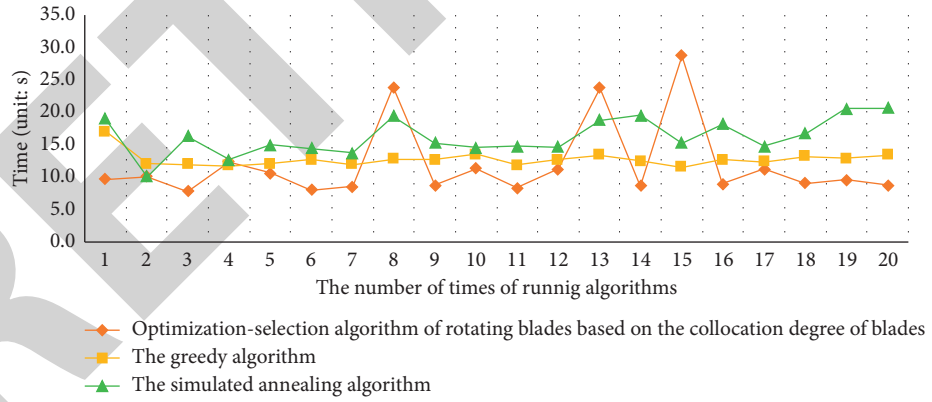


Figure 7: Comparison of solution time of the three algorithms.

the search process, the algorithm may miss the current encountered optimal solution, that is, the optimal solution may be discarded when the simulated annealing algorithm accepts the inferior solution probabilistically, resulting in the final result of simulated annealing algorithm, which is not always the optimal solution.

Step (4) of the intelligent optimization algorithm based on the collocation degree of blades ensures that the algorithm will not destroy the number of groups of blades that have been selected when the algorithm encounters the "bottleneck", which means that when blades are selected by the algorithm, the number of groups of blades already selected and put in the finished product library may increase or maintain the current number, but will not decrease, which is similar to the greedy algorithm; however, the greedy algorithm cannot find the global optimal solution. The simulated annealing algorithm can achieve the global optimal solution by the Metropolis criterion, but at the same time, when it accepts inferior solutions probabilistically, the number of groups of blades that have been

selected and put in the finished product library will decrease, which results in the final solution result of simulated annealing not always being the global optimal solution.

In terms of the solution time, the solution time of all the above three algorithms is not long, and the longest is no more than 30 seconds, which is acceptable in the actual application of enterprises. The termination conditions of the three algorithms set in this paper are: ① when picking out 10 groups of blades (280 blades), terminate the running of the algorithm; ② when the number of iterations is reached, the algorithm terminates. As can be known from Figure 6, when the intelligent optimization algorithm based on the collocation degree of blades picks out 9 groups of blades, the termination conditions of picking out 10 groups of blades is not met; therefore, the algorithm continues to select blades until the number of iterations is reached; as a result, it costs more time. In the other 17 times of running algorithm, when 10 groups of blades are selected, the algorithm will terminate, even if the number of iterations is not used up, so it costs less time. Since the greedy algorithm cannot find the global optimal solution, it terminates only when its number of iterations is used up every time, so the solution time of the greedy algorithm is more stable than the other two algorithms. The solution time of the optimization algorithm based on the collocation degree of blades is not as robust as the other two algorithms, but its solution results are significantly better than the greedy algorithm and the simulated annealing algorithm. In general, within the solution time range acceptable by the enterprise, the optimization algorithm based on the collocation degree of blades has the best solution effect, which can realize efficient selection and full utilization of rotating blades.

5. The Solution and Analysis of Optimizing Matching Problem of Rotor Blades of Aeroengine Fan

5.1. Assembly Sequence Planning of Fan Rotor Blades Based on Simulated Annealing Algorithm. "Optimizing matching" is to plan the assembly sequence of rotating blades and make the remaining unbalance of the rotor as small as possible. After the blade selection is completed, the assembly sequence of the selected rotor blades should be planned. The goal of the assembly sequence planning is to ensure that the remaining unbalance does not exceed the design value to ensure that the rotor static balance meets the design requirements after the rotor blades are assembled according to the planned assembly sequence. Although the remaining unbalance achieved by the current assembly sequence planning technology of the enterprise does not exceed the design value, it is generally too large, as shown in Table 6. Take the first-stage blades of the fan rotor as an example; Table 6 is the comparison result of the remaining unbalance obtained by the 6 kinds of blade sorting methods of the enterprise and design value.

For the n blades of the first-stage fan rotor, the calculation method of the remaining unbalance is shown in formulas (5)–(8).

$$M_x = \sum_{i=1}^{n} M_i \cos\theta_i, \tag{5}$$

$$M_y = \sum_{i=1}^{n} M_i \sin\theta_i, \tag{6}$$

$$M_{\text{left}} = \sqrt{M_x^2 + M_y^2}, \tag{7}$$

$$\alpha = \arctan\frac{M_y}{M_x}. \tag{8}$$

M_x and M_y are the components of the sum of gravitational moments in the x and y directions, respectively; M_i is the gravitational moment of the No. I blade; θ_i is the angle between the gravitational moment vector of the No. I blade and the x-axis; M_{left} is the residual unbalance; and α is the angle of the residual unbalance.

Although the current assembly sequence planning methods of the enterprise can make the remaining unbalance of the rotor blade within the design range, if the remaining unbalance is too large, the remaining unbalance of the rotor will exceed the design value due to errors during the assembly process. Therefore, this article takes the assembly sequence planning of the first-stage blades of the fan rotor as an example, takes the minimum residual unbalance as the goal, and takes the gravitational moment difference of the two blades at the diagonal position of 180°, which does not exceed 1500 g ∗ mm as the constraint (the constraint is specified by the enterprise); the simulated annealing algorithm is used to plan the assembly sequence of the blades, and the obtained residual unbalance by simulated annealing can reach 0.52 g ∗ mm, and the corresponding assembly sequence is shown in Table 7. The running time of SA algorithm is 7.8 seconds. The residual unbalance comparison results of the current assembly sequence planning methods of the enterprise and the assembly sequence planning method based on SA are shown in Table 6. The results show that the assembly sequence planning method based on the simulated annealing algorithm has the advantages of high efficiency and high precision. The flow of the simulated annealing algorithm is shown in Figure 8. The assembly sequence diagram of blades obtained by simulated annealing is shown in Figure 9. The small arrow at the center of Figure 9 indicates the location of the heavy point of the blades. In order to better ensure the static balance of the rotor, when assembling the rotating blades, the location of the heavy point of the rotating blades should be assembled with the location of the light point of the rotor disk edge.

Figure 10 is a statistical chart of the solution accuracy and solution time of running the simulated annealing algorithm for multiple times. As can be seen from Figure 10, the residual unbalance obtained by the simulated annealing algorithm does not exceed 8 g ∗ mm, which is significantly better than the value of 32–83 g ∗ mm (see Table 6)

Table 6: The comparison of the remaining unbalance obtained by 6 kinds of blade sorting methods of the enterprise and design value.

Method 1 2-single beam h/l	Method 2 3-double beam h/l	Method 3 4-quadr beam h/l	Method 4 5-quadr beam h/l	Method 5 6-triple beam h/l	Method 6 7-sequence beam h/l	SA algorithm intelligent sorting	Design value (g*mm)
48	32	40	52	37	83	0.52	100

Table 7: The assembly sequence planned by simulated annealing algorithm (gravitational moment: g*mm).

Assembly sequence	Blade number	Gravitational moment	Assembly sequence	Blade number	Gravitational moment	Assembly sequence	Blade number	Gravitational moment
1	22	272860	11	3	276260	21	21	274840
2	2	274780	12	8	275140	22	5	276100
3	6	276120	13	11	275300	23	17	275760
4	13	275840	14	25	273760	24	28	274120
5	23	276460	15	1	273380	25	14	276180
6	24	275080	16	12	274300	26	19	275180
7	9	274960	17	20	276120	27	10	275420
8	18	275880	18	16	275760	28	26	273580
9	7	275660	19	27	276460			
10	15	274100	20	4	275040			

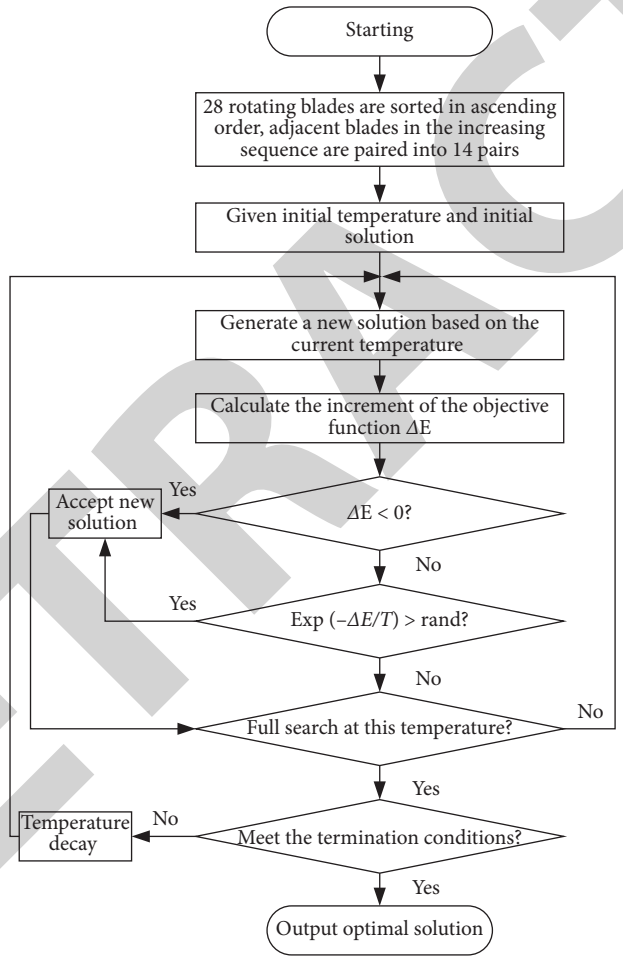


Figure 8: The flowchart of simulated annealing algorithm.

obtained by the enterprise, and is also far less than the design value of 100 g ∗ mm given by the design department. Besides, the running time of the simulated annealing algorithm is between 6 and 12 seconds, and its solution efficiency is very high. Therefore, compared with the current solution methods of the enterprise, the assembly sequence planning technology based on the simulated annealing algorithm not only provides the optimized assembly sequence for the enterprise but also its solution efficiency is very high.

5.2. Assembly Sequence Optimization of the Fan Rotor Blades Based on the Improved Simulated Annealing Algorithm. The simulated annealing (SA) algorithm realizes the global search by accepting inferior solutions probabilistically, so as to achieve the global optimization solution. However, at the same time, precisely because of accepting inferior solutions probabilistically in the search process, the algorithm may miss the current superior solution, and the current superior solution may be the optimal solution of the algorithm, which means the optimal solution may be discarded, and the

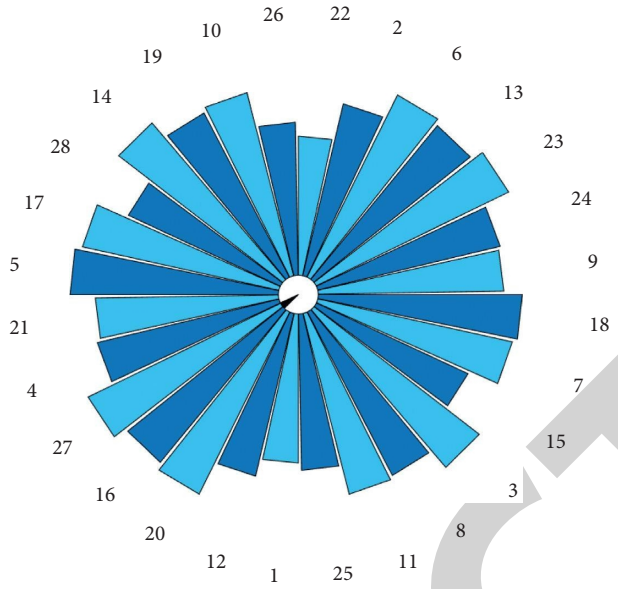


Figure 9: The assembly sequence diagram of blades.

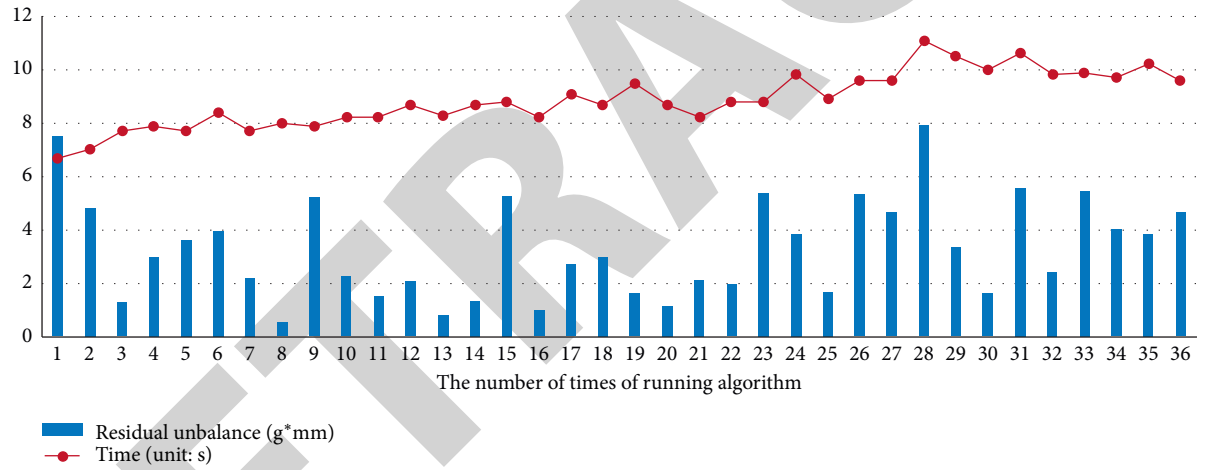


Figure 10: Statistical graph of the solution accuracy and solution time of the simulated annealing algorithm.

optimal solution finally output by the algorithm may not be the true optimal solution. $f = M_{\text{left}} = \sqrt{M_x^2 + M_y^2}$, when $\Delta E = f(\text{new}) - f(\text{old}) < 0$, $f(\text{new})$ is the superior solution, and the inferior solution $f(\text{old})$ will be discarded directly; when $\Delta E = f(\text{new}) - f(\text{old}) > 0$, $f(\text{new})$ is the inferior solution; at this time, turn to the link of accepting inferior solutions probabilistically; if the inferior $f(\text{new})$ is accepted, the superior solution $f(\text{old})$ will be discarded; otherwise, the inferior $f(\text{new})$ will be discarded. In every contest between the superior solution and the inferior solution, there is always a solution discarded; most of the time, the discarded solution is the inferior solution, but at the link of accepting the inferior solution probabilistically, the superior solution may also be discarded. Therefore, the link of accepting the inferior solution probabilistically may cause the algorithm to miss the optimal solution currently encountered. In order to avoid this situation, the SA algorithm is improved and optimized by adding a memory unit to the SA algorithm, and put the discarded old solutions (old solutions include the superior solutions and the inferior solutions) into the memory unit. Finally, an optimal solution will be generated in the memory unit, denoted as MS. Compare the optimal solution MS and the optimal solution (denoted as SAS) obtained by SA algorithm without the memory unit; if MS < SAS, the optimal solution output by the improved simulated annealing algorithm is MS; otherwise, the output is SAS. The flowchart of the improved simulated annealing algorithm is shown in Figure 11.

The comparison results of the solution accuracy and the solution time of the traditional and improved SA algorithms are shown in Figure 12. The solution accuracy is significantly better than the SA algorithm without improvement, but the solution time is slightly slower than the traditional SA algorithm. Running the traditional and improved SA algorithms for 20 times, respectively, and the comparison results of the mean and standard deviation of the solution accuracy

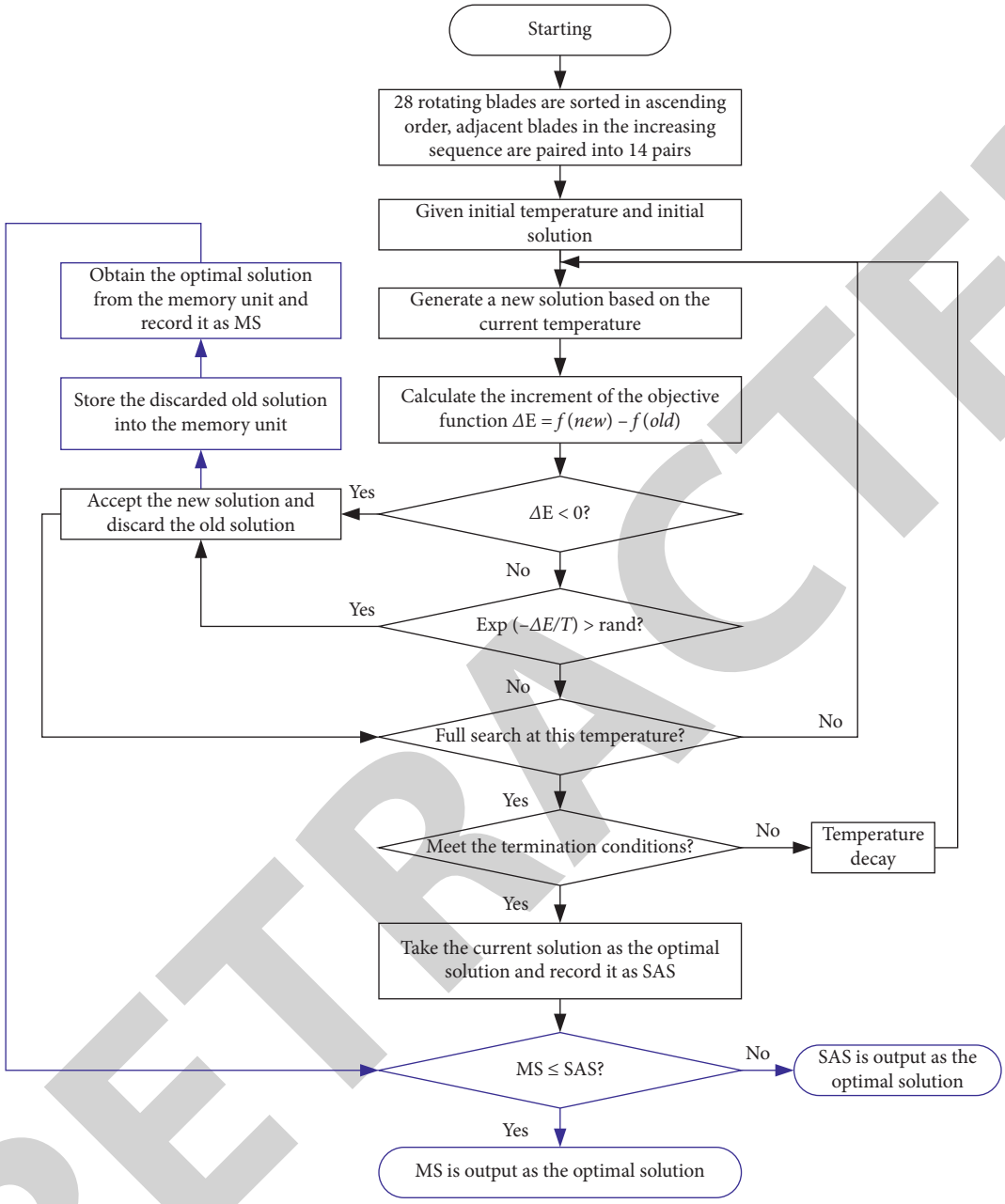


Figure 11: The flowchart of the improved simulated annealing algorithm.

and the solution time are shown in Table 8. The mean and standard deviation of the solution accuracy of the developed algorithm are significantly better than the traditional SA algorithm, but the solution time is a little slower than the traditional SA algorithm. Therefore, the improved simulated annealing algorithm can obtain better solution results, avoiding the situation wherein the algorithm misses the current optimal solution because of accepting the inferior solution probabilistically during the solution process. Moreover, after the algorithm is improved, its average running time is increased only by 2 seconds. Therefore, after the algorithm is improved and optimized, the algorithm's solution accuracy and robustness are improved significantly. Besides, the algorithm's solution efficiency is still very high, which indicates that the improvement and optimization of the simulated annealing not only maintains its efficient solution efficiency but also significantly improves its solution accuracy and robustness.

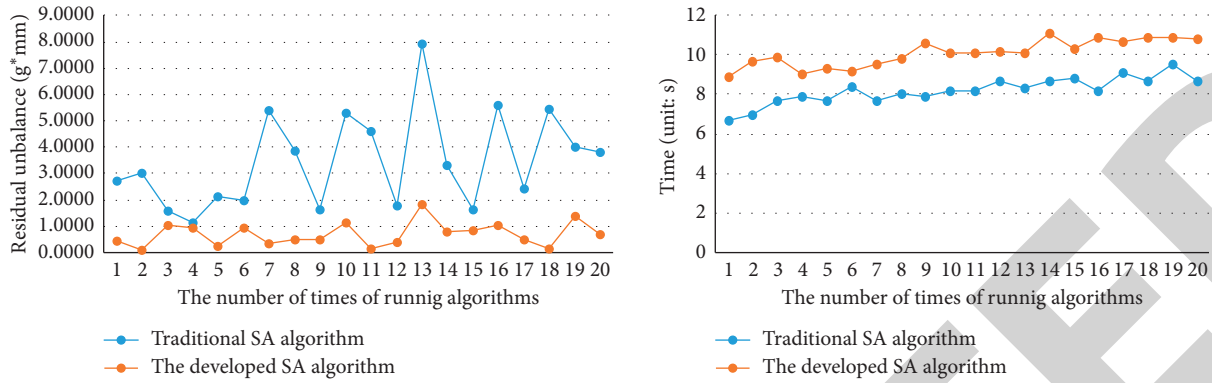


Figure 12: Comparison of solution accuracy and solution time of the traditional SA algorithm and improved SA algorithm.

Table 8: The comparison results of the solution effect of the traditional SA and improved SA.

	Solution accuracy of traditional SA	Solution accuracy of improved SA	Solution time of traditional SA	Solution time of improved SA
Mean	3.4682 g*mm	0.6950 g*mm	8.2 s	10.1 s
Standard deviation	1.7538	0.4412	0.6546	0.6641

6. Conclusions

(1) This paper established the optimization algorithm of selecting blades based on the collocation degree of blades and realized efficient selection and full utilization of rotating blades before assembly. The blade resource utilization rate is increased from 65%–74% currently achieved by the enterprise to 83%–93% by the algorithm. The statistic results of the algorithm running 20 times show that the minimum solution time is 7.7 seconds, the maximum is 28.7 seconds, and the solution efficiency is very high. Moreover, the solution results of the optimization algorithm based on the collocation degree of blades are far ahead of the greedy algorithm and simulated annealing algorithm. Consequently, the algorithm of selecting blades based on the collocation degree of blades realizes the efficient selection and full utilization of blades.

The optimization algorithm of selecting rotating blades based on the collocation degree of blades has two highlights: ① the collocation degree of blades prompts the algorithm to select more blades faster, which ensures the efficient selection of blades; ② the "bottleneck" processing method of the algorithm determines that when the algorithm selects blades, it will not destroy the number of blades that have already been selected, which means the number of blades that have been picked out and put in the finished product library may increase, may maintain the current number, but not decrease, and this highlight ensures that the algorithm can make full use of blade resources.

(2) This paper uses simulated annealing algorithm to plan the assembly sequence of blades, and the residual unbalance obtained by SA algorithm can reach 0.52 g ∗ mm, which is far less than the given design value of 100 g ∗ mm, and 0.52 g ∗ mm is also far smaller than the remaining unbalance (between 30 g ∗ mm and 100 g ∗ mm) achieved by the company's current assembly sequence planning technology. Therefore, the assembly sequence planning method of blades based on simulated annealing algorithm greatly improves the static balance reliability of rotors.

The improved simulated annealing algorithm not only maintains the characteristic of efficient solution but also improves the solution accuracy and solution robustness significantly. Consequently, the improved simulated annealing algorithm provides optimized assembly sequence and efficient solution method for the assembly of rotor blades.

Data Availability

All the data and materials of the manuscript are available, and they are from the National Key Research and Development Program of China. The data can be provided in Supplementary Information files that are submitted along with the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interests.

Acknowledgments

The authors gratefully acknowledge the financial support by the National Key Research and Development Program of China under Grant no. 2019YFB1703800.

Hindawi
Shock and Vibration
Volume 2021, Article ID 6694896, 19 pages
https://doi.org/10.1155/2021/6694896

Research Article

A New Maintenance Optimization Model Based on Three-Stage Time Delay for Series Intelligent System with Intermediate Buffer

Xiaolei Lv,[1] **Qinming Liu**,[1] **Zhinan Li**,[1] **Yifan Dong**,[2] **Tangbin Xia**,[2] **and Xiang Chen**[1]

[1]*Department of Industrial Engineering, Business School, University of Shanghai for Science and Technology, 516 Jungong Road, Shanghai 200093, China*
[2]*State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, SJTU-Fraunhofer Center, Shanghai 200240, China*

Correspondence should be addressed to Qinming Liu; lqm0531@163.com

Received 14 December 2020; Revised 7 January 2021; Accepted 12 February 2021; Published 26 February 2021

Academic Editor: Vasudevan Rajamohan

For the maintenance problem of intelligent series system with buffer stock, a preventive maintenance model based on the threestage time delay theory is proposed. Firstly, the intelligent series system is decomposed into $n-1$ virtual series systems by using approximate decomposition method. The impact factor is introduced to establish the failure rate and maintenance rate model of each virtual machine. Secondly, a preventive maintenance model based on the three-stage time delay theory is proposed for each virtual series system. The machine state from normal operation to failure stage is divided into three steps: initial defect, serious defect, and fault, and different distribution functions are defined in different stages to simulate the degradation process of the machine. Based on the three-stage time delay theory, the machine cost ratio model was established by taking the machine monitoring time and buffer stock as decision variables and the minimum unit time cost rate as objective function. Finally, the rationality and validity of the model are verified by an example analysis, which provides a basis for the maintenance of the intelligent series system.

1. Introduction

Intelligent series system is an important part of modern industrial manufacturing system. Due to the variety of machines and complex layout and structure, any failure may lead to the shutdown of the entire production line and cause huge economic losses for enterprises. For the continuous series production line, the preventive maintenance can also cause downtime.

Reasonably adding buffer between two machines can improve the flexibility of the production line, reduce the production dependence between upstream and downstream machine, and reduce the impact on the stability of series system due to machine downtime. The performance of the intelligent series system is closely related to its preventive maintenance plan and buffer setting. Preventive maintenance is related to the buffer stock allocation. In order to improve the production line stability and reduce the cost, it is very necessary to jointly optimize the series system buffer stock allocation and maintenance plan of machines.

Machines in the intelligent series system are closely connected, and the failure of one machine will lead to the shutdown of the entire intelligent series system. Recently, there are many literature works on the optimal preventive maintenance strategy of the series system. Wu et al. [1] established an optimized maintenance cost model and determined the optimal condition monitoring interval and the degradation level after imperfect preventive maintenance. The authors in [2] developed a dynamic maintenance strategy joint optimization problem that integrates production and opportunity maintenance. Rooeinfar et al. [3] studied the scheduling problem of uncertain flexible pipeline with finite buffer. Zhang et al. [4] investigated the incorporation of balance and preventive maintenance in U-shaped assembly line. Two metaheuristic algorithms including elitist nondominated sorting genetic algorithm and

multiobjective simulated annealing algorithm were designed to solve this problem. The authors in [5] studied the integrated control of dynamic maintenance and production in a deteriorating manufacturing system and proposed a dynamic maintenance strategy that included corrective, preventive, and opportunistic maintenance. Bouslah et al. [6] discussed the integrated production, quality, and maintenance control of the production line. Motlagh et al. [7] developed an expert system for the unreliable unbalanced production line in reality, in which all time-based parameters were random. Considering a series production system with random degradation, Wang et al. [8] proposed a predictive maintenance strategy based on the predicted failure probability of each machine and a production control strategy based on the target service level, so as to meet the dynamic stochastic demand in each period. Wang et al. [9] proposed two maintenance strategies for the series production line. The first was based on the cost rate of the machine under long-term operation, and the second was based on the single-piece machine maintenance strategy considered in the production line. The authors in [10] proposed an alternative scheduling model for railway production lines and proposed a time-based flexible displacement (FTBR) method in combination with the artificial bee colony (ABC) algorithm. Based on the concept of "energy-saving opportunity window," [11] modeled the continuous deterioration process of each machine and regarded the energy-saving opportunity window of the production system as the opportunity window of preventive maintenance. However, the above literature only proposes preventive maintenance strategies for series system not combined with buffer zones.

Buffers have been used in maintenance of machines for a long time. The authors in [12] first proposed buffer stock, considering the impact of the interstage buffer on production at different production speeds, different failure rates, and different repair rates. The method of using regeneration point for analysis and treatment was given. Currently, a large number of scholars study the optimal allocation of buffer. The authors in [13] presents a model to determine the optimal length of continuous production periods between maintenance actions and the optimal buffer inventory to satisfy demand during preventive maintenance or repair of a manufacturing facility. On the basis of this model, the authors in[14] considers that the opportunities for the fabrication of the buffer inventory and opportunities to carry out a maintenance action to the production facility are random. The authors in [15] proposed a multiobjective mathematical formula and hybrid method, which can simultaneously solve the buffer size and machine allocation problems on unreliable production lines and assembly lines with general distributed time-varying parameters. The authors in [16] proposed a tabu search algorithm to find the optimal buffer allocation plan for a serial production line composed of unreliable machines. The authors in [17] considered an imperfect production system with preventive maintenance activities in order to obtain the optimal buffer stock and minimum warranty inspection policy for sold products. The production system has the probability of changing from the normal state to the out-of-control state at any time. Under the normal state and out-of-control state, the production system will produce a certain proportion of defect items. The authors in [18] developed a method to analyze the complex tradeoff between the preventive maintenance and the buffer's contribution to system performance, considering a two-machine continuous manufacturing system with a finite capacity buffer. The authors in [19] analyzed the tradeoff between buffer capacity, spare parts inventory, and throughput for a two-stage production system with buffers and established a discrete-time Markov chain for two different situations. The numerical examples showed that the effect of a spare part on the efficiency of a transfer line was much greater than the effect of additional buffer places. All of the above research is aimed at two-stage system and does not combine buffer with intelligent series system.

There are few studies on the combination of maintenance and buffer with series system. Nahas [20] considered an unreliable serial production line. The target was to minimize the total cost of the system through finding the optimal preventive maintenance strategy and optimal buffer size at a given level of the system throughput. Extend the flood algorithm was put forward in order to solve this problem. The decomposition approximation method was used to estimate the production capacity of production line. Zandieh et al. [21] studied buffer and preventive maintenance cycle allocation issues. The model was built with three objective functions: the maximization of production rate, the minimization of buffer size, and the total number of defect units. Finally, a synthetic simulation method and a metaheuristic algorithm were used to solve the model. Lopes [22] minimized the total cost of each product while considering product quality testing for production lines with buffers. According to the machine degradation stage and buffer level, Kang and Ju [23] used Markov decision model to obtain the optimal maintenance strategy of the machine in the production line. Alfieri et al. [24] used the approximated mathematical programming formulation of the Buffer Allocation Problem (BAP) simulation-optimization based on the time buffer concept. However, in these studies, the preventive maintenance strategy of the series system is considered as a whole, and it is not disassembled to study each machine in the production line. Preventive maintenance strategy may not be adapted to each machine in the production line.

In this paper, the buffer is added to the intelligent series system to jointly optimize the buffer stock and the optimal preventive maintenance cycle. The approximate decomposition method is used in this paper to decompose the intelligent series system into several virtual series systems with two machines and a buffer.

The approximate decomposition method is used to study the problem of production line preventive maintenance, which was first proposed by [25]. After that, it was widely used. Based on the approximate decomposition method, [26] presented an efficient method to evaluate performance of tree-structured assembly/disassembly (AD) systems with finite buffer capacity. For mixed-model flexible transfer lines, [27] proposed a general simulation model. Compared with

the approximate decomposition method, the numerical results show that the proposed method is robust for predicting the throughput of transmission lines. Li et al. [28] proposed a common model that unifies several approximate methods for the analysis of tandem queueing systems with blocking. Xia et al. [29] proposed an efficient decomposition method based on a generalized exponential distribution to analyze the homogeneous transfer lines with unreliable buffers. The authors in [30] developed three heuristic approaches to solve the formulated combinatorial optimization problem. To estimate the production line throughput, an approximate decomposition method was used. Xia et al. [31] decomposed the original long line into several small decoupled subsystems and added relation condition variables between the subsystems. Bai et al. [32] proposed a new aggregation-based iterative algorithm to calculate the performance metrics of a multimachine serial line by representing it using a group of virtual two machine lines. In this paper, based on the approximate decomposition method, the influence factors are introduced according to the importance of different machines in the series system to obtain the optimal maintenance strategy for each machine.

Another innovation of this paper is to introduce the time delay theory into the maintenance model. Many studies have presented the preventive maintenance strategies by considering buffer stock. However, the machine's degradation is not considered in this area. In this paper, the preventive maintenance strategy can be developed by introducing the time delay theory to simulate the degradation process.

The time delay theory is often used to simulate the machine degradation process. The time delay model proposed by Christer and Waller [33] is the first time to extend the time delay theory to the maintenance of industrial plants. The basic model of inspection and maintenance and some change models observed in practice are presented. Later, a large number of scholars applied the time delay theory to the field of establishing the correlation between machine maintenance cost and preventive maintenance inspection interval cycle. Wang [34] proposed such a model for a serviceable one-component system to jointly model the effect of RS and inspection with replacement on the basis of the delay-time concept. Zhao et al. [35] developed a model to evaluate the reliability and optimized the inspection schedule for a multidefect component. Gomes da Silva and Lopes [36] simulated the preventive maintenance model based on the nonhomogeneous Poisson distribution. Mahmoudi et al. [37] studied the occurrence process of machine defects presented as homogeneous Poisson distribution, and then solved the optimal maintenance cycle of preventive maintenance strategy. Based on the traditional time delay theory, Wang et al. [38] introduces a two-level inspection policy model for a single component plant system based on a three-stage failure process. They divided the machine failure into three states: original defect, serious defect, and fault, so as to simulate the fault random process. By applying the three-stage time delay theory to the simulation of machine degradation and renewal process, the clustering summary of different kinds of faults or defects that may occur in the machine can be performed in a more precise and quantitative manner.

In this paper, first, the intelligent series system is decomposed into $n-1$ virtual series systems by approximate decomposition method, and one virtual series system includes two virtual machines. The aim is to find the relationship between machines and present the cost ratio and maintenance ratio model by considering an influence factor. Then, for each two virtual machines, the buffer stock and machine monitoring time can be described as decision variables, and the total cost rate can be minimized as the objective function. A novel maintenance model based on three-stage time delay is developed to obtain the optimal preventive maintenance strategy and the buffer stock. The proposed model can be divided into four types, and it contains the whole process of machine degradation based on different machines status and monitoring time. Finally, the proposed model is compared with the maintenance model based on two-stage time delay by a case study. The overall optimal maintenance strategy and buffer stock are obtained. In the case study, this paper analyzes an intelligent series system of Shanghai Pangyuan Machinery Co.

The remainder of the paper is as follows. In Section 2, a description of general problem is presented. In Section 3, the notation and assumption are presented. The maintenance rate and cost ratio model of virtual machine and the preventive maintenance model based on the three-stage time delay theory are introduced in Section 4. In Section 5, solving method is introduced. In Section 6, numerical examples are presented and analyzed. Finally, the most important results and future work are summarized in Section 7.

2. Problem Description

The intelligent series system L includes n machines and $n-1$ buffers. It is shown in Figure 1. We assume that M_1 is never starved and the supply of raw materials needed by M_1 is continuous. M_n is never blocked, and the final product produced by M_n does not have a backlog state. The failure rate λ_i and maintenance rate μ_i of each unit are known and defined as known conditions. The approximate decomposition method is used to decompose L into $n-1$ virtual series systems, and each virtual series system includes two machines and one buffer.

After L is decomposed, $n-1$ virtual series systems are formed. M_u and M_d are virtual machines of one virtual series system after decomposition. M_u is the upstream machine, and semifinished product m_1 from M_u is input into downstream machine M_d at the production rate β through the intermediate buffer B. M_d uses m_1 as the raw material to produce product m_2 at β. B needs to be accumulated before maintenance actions adopted by M_u to ensure continuity of production process of the virtual series system.

The traditional time delay model divides the health status into defect state and fault state. The three-stage time delay model divides the health status into normal state, original defect state, serious defect state, and fault state, as shown in Figure 2. Thus, the degradation process can be divided into original defect time, serious defect time, and failure time.

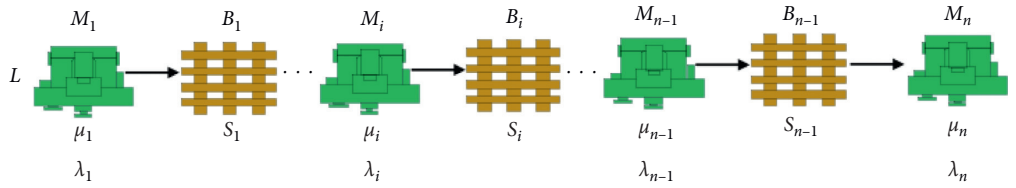


Figure 1: Intelligent series system L.

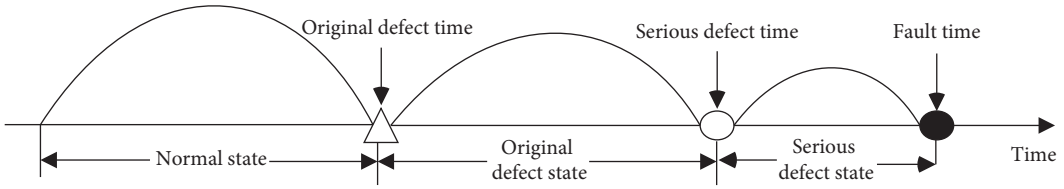


Figure 2: Three stages of failure.

After the upstream machine runs a certain period, B can be added by the replenishment rate α until it reaches buffer stock level S. Then, status monitoring can be executed on the upstream machine. If the monitoring result shows that it is in initial defect state or serious defect state, preventive maintenance needs to be adopted immediately. If it is stopped due to failure before monitoring, then fault repair can be performed.

3. Notation and Assumption

3.1. Notation

M_i: i^{th} machine of intelligent series system L

λ_i: failure rate of M_i

μ_i: maintenance rate of M_i

$M_u(i)$: upstream machine after decomposition

$\lambda_u(i)$: failure rate of $M_u(i)$

$\mu_u(i)$: maintenance rate of $M_u(i)$

$M_d(i)$: downstream machine after decomposition

$\lambda_d(i)$: failure rate of $M_d(i)$

$\mu_d(i)$: maintenance rate of $M_d(i)$

$t_u(i)$: maintenance time of $M_u(i)$

$r_u(i)$: remaining maintenance time of $M_u(i)$

α: replenishment rate of buffer

β: production rate of L

X: machine normal operation phase

Y: machine original defect operation phase

Z: machine serious defect operation phase

$f_x(x)$: probability density function from initial state of one machine to occurrence of original defect

$f_y(y)$: probability density function from original defect state of one machine to the occurrence of serious defect

$f_z(z)$: probability density function from serious defect state of one machine to the occurrence of failure

$F_x(x)$: distribution function from initial state of one machine to the occurrence of original defect

$F_y(y)$: distribution function from original defect state of one machine to the occurrence of serious defect

$F_z(z)$: distribution function from serious defect state of one machine to the occurrence of failure

h: buffer stock holding cost per unit

ρ: shortage cost per unit

C_r: monitoring cost of each time

C_x: maintenance cost of one time for machine in an original defect state

C_y: maintenance cost of one time for machine in a serious defect state

C_z: maintenance cost of one time for machine in a fault state

$C_m(T)$: maintenance costs in a cycle

$C_h(S)$: holding cost in a cycle

$C_s(S,T)$: shortage cost in a cycle

S: buffer stock level

T: machine monitoring time from the end of a maintenance action to the start of the status monitoring

EC(T): expected length of a cycle for operational time T

$C(S,T)$: total cost in a cycle

TCR (S,T): cost rate in a cycle

W_i $(i = 1,2,3)$: random variable, maintenance time of the machine by original defect state, serious defect state, and fault state, respectively

$g_i(t)$: probability density function of W_i, $i = 1,2,3$

$G_i(t)$: distribution function of W_i, $i = 1,2,3$

3.2. Assumptions

(1) The machine status needs to be monitored after T, and monitoring time can be ignored.

(2) All the states can be accurately monitored. The corresponding maintenance actions can be adopted immediately after monitoring. Production can be resumed immediately after completing maintenance.

(3) Extra production capacity is always available in order to produce buffer stock.

(4) If there is buffer stock after completing maintenance, the next production cycle firstly consumes buffer stock.

4. The Model

4.1. Decomposing Production Line. L is decomposed into $n-1$ virtual series systems by approximate decomposition method, as shown in Figure 3. Each virtual series system has only two virtual machines and a buffer (where M_u is never starved and M_d is never blocked). Buffer in Line 1 corresponds to B_1, and buffer in Line $n-1$ corresponds to B_{n-1}. For Line i, failure rates $\lambda_u(i)$, $\lambda_d(i)$ and maintenance rates $\mu_u(i)$, $\mu_d(i)$ are unknown. Thus, the next step is to solve the failure rate and maintenance rate of virtual machine.

4.2. Failure Rate and Maintenance Rate Model of Virtual Machine. M_i in L is decomposed into virtual machine $M_d(i-1)$ and $M_u(i)$, $M_u(i)$, where $M_u(i)$ is upstream machine of Line i and $M_d(i-1)$ is downstream machine of Line $i-1$. Maintenance strategy of $M_u(i)$ is the same as M_i in L. The starvation of M_i represents starvation of $M_d(i-1)$, and failure represents failure of $M_u(i)$. Thus, λ_u and μ_u of $M_u(i)$ are required, and its maintenance strategy is formulated. $M_u(i)$ is a virtual machine decomposed from M_i. The failure of $M_u(i)$ represents failure or starvation of M_i, M_i, and the starvation of M_i is caused by failure or starvation of M_{i-1}. Failure or starvation of M_{i-1} represents failure of $M_u(i-1)$. Thus, the failure $M_u(i)$ is jointly determined by failure of M_i and failure of $M_u(i-1)$.

The failure of $M_u(i)$ is related to failure of $M_u(i-1)$ and failure of M_i. Thus, the impact factor a can be introduced. The proportion is a when the failure of $M_u(i)$ is from $M_u(i-1)$fault, and the proportion is $1-a$ when the failure of $M_u(i)$ is from M_i fault. The relationship is as follows:

$$\lambda_u(i) = a\lambda_u(i-1) + (1-a)\lambda_i, \tag{1}$$

$$t_u(i) = ar_u(i-1) + (1-a)t_i, \tag{2}$$

where $t_u(i)$ and t_i are the average maintenance time of $M_u(i)$ and M_i respectively. $r_u(i-1)$ is the average remaining maintenance time of $M_u(i-1)$ when M_i is starved. It is the maintenance time after consuming inventory in B_{i-1}.

4.3. Cost Ratio Model. For each cycle, the machine can be monitored immediately after completing buffer stock replenishment. Different maintenance strategies can be adopted based on machine monitoring status. The possible occurrence time of original defect state, serious defect state, and fault state is T_x, T_y, T_z, respectively. There are four different situations for adopting preventive maintenance based on monitoring status, T and S.

4.3.1. $0 < T < T_x$. The state monitoring time of machine occurs before the original defect time. Machine status is in a nondefective state, and it is unnecessary to execute any maintenance action. After that, in order to prevent the failure and to detect the defect in time, it is necessary to execute a state monitoring every day until the original defect is detected. Buffer stock has been replenished from first status monitoring. Buffer stock level change during a running cycle is shown in Figure 4 and preventive maintenance of machine is executed under the original defect state.

(1) The probability of Twithin $[0, T_x]$:

$$P(0 < T < T_x) = P^1 = \int_T^{\infty} f_x(x)\mathrm{d}x. \tag{3}$$

(2) Operation cycle of machine for $0 < T < T_x$

The operational cycle of machine includes monitoring time and maintenance time. One cycle is from the end of the above maintenance to the end of the next maintenance. For this situation, operational time is T_x, and maintenance time is W_1. Then,

$$\mathrm{EC}^1(T) = E(W_1) + E(T_X). \tag{4}$$

The total cost includes inventory holding cost, shortage cost, and maintenance cost within a cycle.

(3) Inventory holding cost for $0 < T < T_x$

The inventory holding cost can be generated from beginning to produce buffer stocks, and it can be increased with the increasing of buffer stock. From the beginning of buffer stock replenishment to the end of maintenance actions, buffer is always occupied. Thus,

$$C_h^1(S) = h\left[\frac{S^2}{2}\left(\frac{\alpha+\beta}{\alpha\beta}\right) + (x-T)S\right]. \tag{5}$$

(4) Shortage cost for $0 < T < T_x$

A shortage occurs when buffer stocks are depleted and maintenance actions have not ended. Therefore, there will be a shortage of cost. The shortage time is from the end of buffer stock depletion to the end of maintenance action. Then,

$$C_s^1(S,T) = \rho\beta \int_{S/\beta}^{\infty} \overline{G_1}(w)\mathrm{d}w. \tag{6}$$

(5) Maintenance cost for $0 < T < T_x$

Maintenance cost includes the expected cost of preventive maintenance and all testing costs. In this case, preventive maintenance of the original defect state is carried out. $x - T + 1$ tests were conducted before the maintenance activity. Thus, maintenance cost in a cycle is

$$C_m^1(T) = C_x + (x - T + 1)C_r. \tag{7}$$

The expected cost in a cycle for $0 < T < T_x$ is the summation of inventory holding cost, shortage cost, and maintenance cost. The expected cost $C^1(S,T)$is obtained:

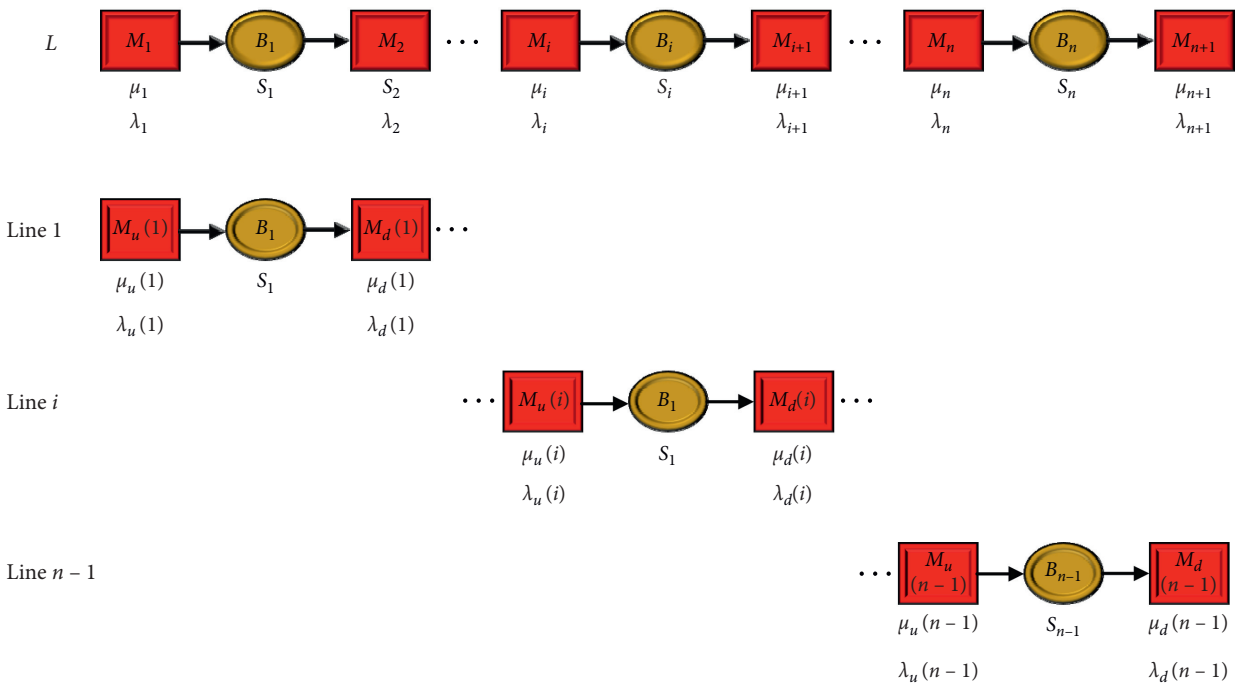


Figure 3: $n-1$ virtual series systems after decomposition.

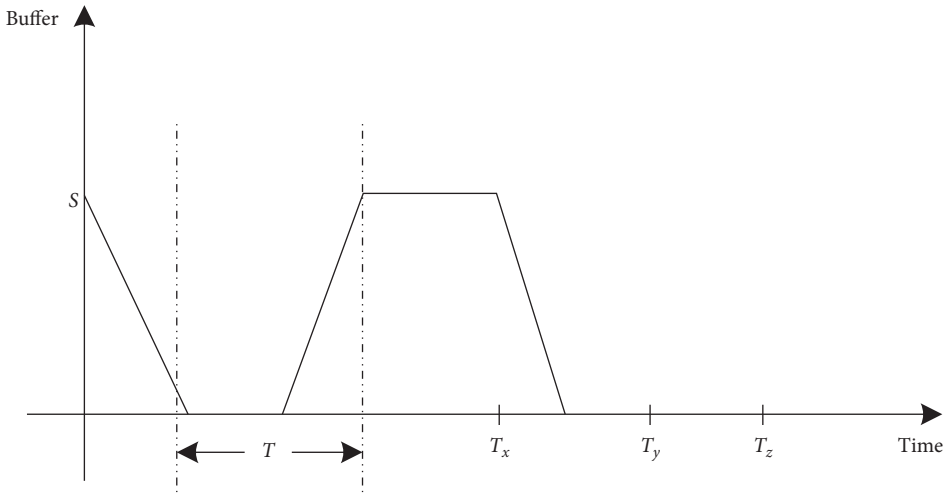


Figure 4: Buffer stock change diagram in a cycle for $0 < T < T_x$.

$$C^1(S,T) = C_h^1(S) + C_s^1(S,T) + C_m^1(T)$$
$$= h\left[\frac{S^2}{2}\left(\frac{\alpha+\beta}{\alpha\beta}\right) + (x-T)S\right] + \rho\beta\int_{S/\beta}^{\infty}\overline{G_1}(w)\mathrm{d}w + C_x + (x-T+1)C_r. \quad (8)$$

4.3.2. $T_x < T < T_y$. The state monitoring time of machine occurs after the original defect time and before the serious defect time. Machine status is in an original defect operation state, and it needs to execute preventive maintenance of the original defect state. After the buffer stock is replenished, machine status is monitored immediately. Buffer stock level change during a running cycle is shown in Figure 5.

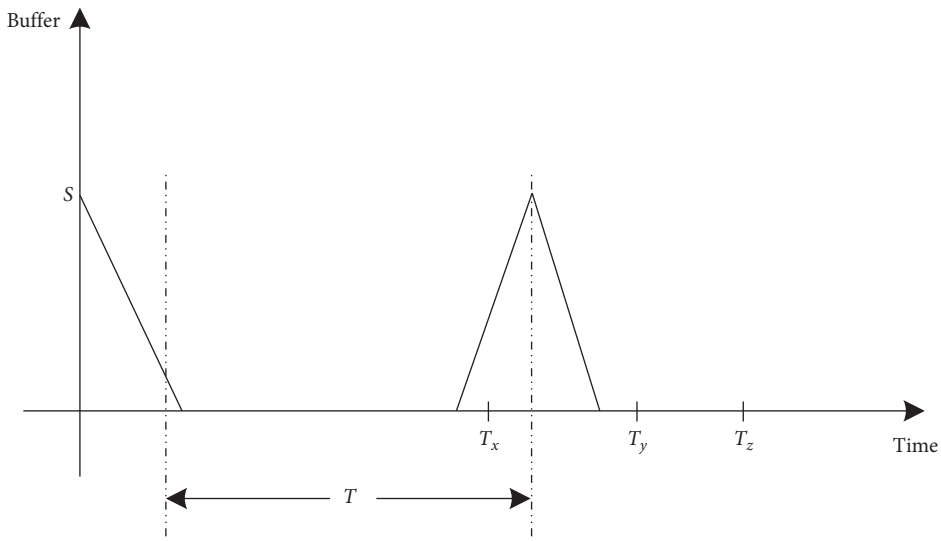


Figure 5: Buffer stock change diagram in a cycle for $T_x < T < T_y$.

(1) The probability of T within $[T_x, T_y]$

$$P\left(T_x < T < T_y\right) = P^2 = \int_0^T \int_{T-x}^{\infty} f_x(x) f_y(y) \mathrm{d}x\mathrm{d}y = \int_0^T f_x(x)\left(1 - F_y(T-x)\right)\mathrm{d}x. \tag{9}$$

(2) Operation cycle of machine for $T_x < T < T_y$

The operational cycle of machine includes operational time and maintenance time. One cycle is from the end of the above maintenance to the end of the next maintenance. For this situation, operational time of machine is T, and maintenance time is W_1. Then,

$$\mathrm{EC}^2(T) = E\left(W_1\right) + T. \tag{10}$$

(3) Inventory holding cost for $T_x < T < T_y$

The inventory holding cost can be generated from beginning to produce buffer stocks, and it can be increased with the increasing of buffer stock. From the beginning of buffer stock replenishment to the end of maintenance actions, buffer is always occupied. Thus,

$$C_h^2(S) = h\left[\frac{S^2}{2}\left(\frac{\alpha+\beta}{\alpha\beta}\right)\right]. \tag{11}$$

(4) Shortage cost for $T_x < T < T_y$

The shortage cost in one cycle is the same as in Section 4.3.1.

$$C_s^2(S,T) = \rho\beta \int_{S/\beta}^{\infty} \overline{G_1}(w)\mathrm{d}w. \tag{12}$$

(5) Maintenance cost for $T_x < T < T_y$

In this case, the machine makes one state monitoring process. The test result is the original defect state, so the preventive maintenance of the original defect state is executed. Thus, maintenance cost in a cycle is

$$C_m^2(T) = C_x + C_r. \tag{13}$$

The expected cost $C^2(S,T)$ in a cycle for $T_x < T < T_y$ is obtained as

$$\begin{aligned} C^2(S,T) &= C_h^2(S) + C_s^2(S,T) + C_m^2(T) \\ &= h\left[\frac{S^2}{2}\left(\frac{\alpha+\beta}{\alpha\beta}\right)\right] + \rho\beta \int_{S/\beta}^{\infty} \overline{G_1}(w)\mathrm{d}w + C_x + C_r. \end{aligned} \tag{14}$$

4.3.3. $T_y < T < T_z$. The state monitoring time of machine occurs after the serious defect time and before the failure time. Machine status is in a serious defect operation state, and it needs to execute preventive maintenance of the serious defect state. After the buffer stock is replenished, the machine status is monitored immediately. Buffer stock level change during a running cycle is shown in Figure 6.

(1) The probability of Twithin $[T_y, T_z]$

$$\begin{aligned} P\left(T_y < T < T_z\right) = P^3 &= \int_0^T \int_0^{T-x} \int_{T-x-y}^{\infty} f_x(x) f_y(y) f_z(z)\mathrm{d}x\mathrm{d}y \\ &= \int_0^T \int_0^{T-x} f_x(x) f_y(y)\left(1 - F_z(T-x-y)\right)\mathrm{d}x\mathrm{d}y. \end{aligned} \tag{15}$$

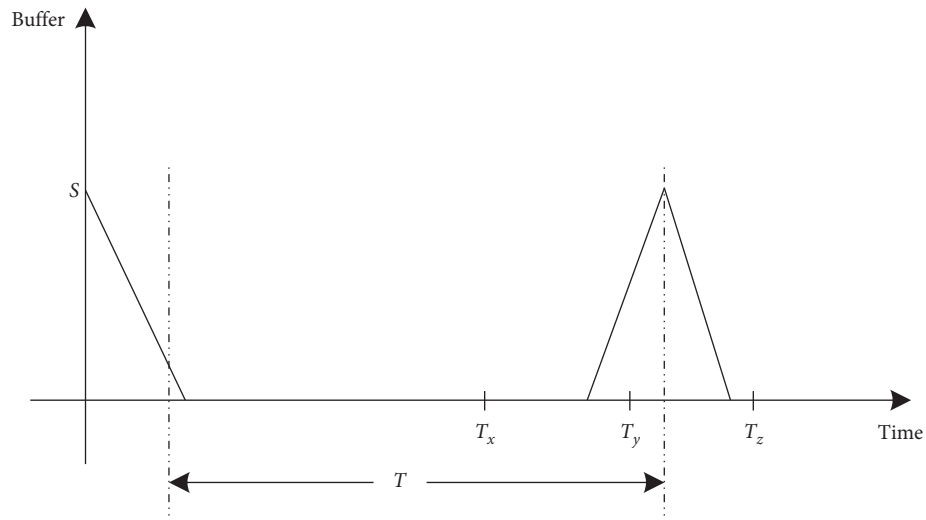


Figure 6: Buffer stock change diagram in a cycle for $T_y < T < T_z$.

(2) Operation cycle of machine for $T_y < T < T_z$

For this situation, operational time of machine is T, and maintenance time is W_2. Then,

$$\mathrm{EC}^3(T) = E(W_2) + T. \tag{16}$$

(3) Inventory holding cost for $T_y < T < T_z$

For this situation, from the beginning of buffer stock replenishment to the end of maintenance activities, there will be inventory occupation. Therefore, the inventory holding cost is as follows:

$$C_h^3(S) = h\left[\frac{S^2}{2}\left(\frac{\alpha+\beta}{\alpha\beta}\right)\right]. \tag{17}$$

(4) Shortage cost for $T_y < T < T_z$

In this case, preventive maintenance of machine in the serious defect state is required. Thus, the shortage cost in a cycle is

$$C_s^3(S,T) = \rho\beta\int_{S/\beta}^{\infty}\overline{G_2}(w)\mathrm{d}w. \tag{18}$$

(5) Maintenance cost for $T_y < T < T_z$

In this case, the machine makes one state monitoring process. The test result is the serious defect state, so the preventive maintenance of the serious defect state is carried out. Thus, maintenance cost in a cycle is

$$C_m^3(T) = C_y + C_r. \tag{19}$$

The expected cost $C^3(S,T)$ in a cycle for $T_y < T < T_z$ is obtained.

$$\begin{aligned} C^3(S,T) &= C_h^3(S) + C_s^3(S,T) + C_m^3(T) \\ &= h\left[\frac{S^2}{2}\left(\frac{\alpha+\beta}{\alpha\beta}\right)\right] + \rho\beta\int_{S/\beta}^{\infty}\overline{G_2}(w)\mathrm{d}w + C_y + C_r. \end{aligned} \tag{20}$$

4.3.4. $T > T_z$. In this case, the machine fails before the state detection is carried out. The planned state detection takes place after the failure of machine. At this time, the corrective maintenance action is executed. Buffer stock has not been replenished, or the production of buffer stock has not started before the failure shutdown. Thus, the buffer stock change during a running cycle is shown in Figure 7.

(1) The probability of T within $[T_z, \infty]$

$$\begin{aligned} P(T > T_z) = P^4 &= \int_0^T\int_0^{T-x}\int_0^{T-x-y} f_x(x)f_y(y)f_z(z)\mathrm{d}x\mathrm{d}y\mathrm{d}z \\ &= \int_0^T\int_0^{T-x} f_x(x)f_y(y)F_z(T-x-y)\mathrm{d}x\mathrm{d}y. \end{aligned} \tag{21}$$

(2) Operation cycle of machine for $T > T_z$

For this situation, before machine status is monitored, the failure has already occurred. Operational time of machine is T_z, and maintenance time is W_3. Then,

$$\mathrm{EC}^4(T) = E(W_3) + E(T_z). \tag{22}$$

(3) Inventory holding cost for $T > T_z$

In this case, from the beginning of buffer stock replenishment to the failure occurrence, there will be inventory occupation. Then,

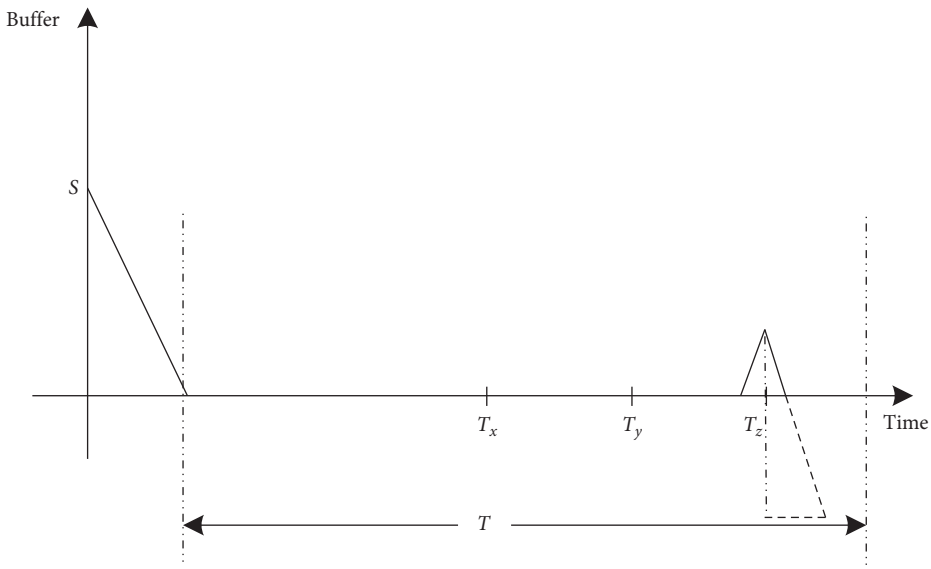


Figure 7: Buffer stock change diagram in a cycle for $T > T_z$.

$$C_h^4(S) = \frac{h}{2}\left[S - \alpha(T - E(T_z))\right]\left[\frac{(\alpha+\beta)S}{\alpha\beta} - \left(\frac{\alpha+\beta}{\beta}\right)(T - E(T_z))\right]. \tag{23}$$

(4) Shortage cost for $T > T_z$

In this case, the machine fails before the buffer stock is replenished. The machine breaks down, so corrective maintenance is carried out. Thus, the shortage cost in a cycle is

$$C_s^4(S,T) = \rho\beta \int_{(S-\alpha(T-E(T_z)))/\beta}^{\infty} \overline{G_3}(w)\mathrm{d}w. \tag{24}$$

(5) Maintenance cost for $T > T_z$

In this case, the status of the machine has not been detected, and the machine breaks down. Therefore, corrective maintenance is executed. Thus, the maintenance cost of machine in a cycle is the maintenance cost under the state of failure, and there is no detection cost. Then,

$$C_m^4(T) = C_z. \tag{25}$$

The expected cost $C^4(S,T)$ in a cycle for $T > T_z$ is obtained:

$$\begin{aligned} C^4(S,T) &= C_h^4(S) + C_s^4(S,T) + C_m^4(T) \\ &= \frac{h}{2}\left[S - \alpha(T - E(T_z))\right]\left[\frac{(\alpha+\beta)S}{\alpha\beta} - \left(\frac{\alpha+\beta}{\beta}\right)(T - E(T_z))\right] \\ &\quad + \rho\beta \int_{(S-\alpha(T-E(T_z)))/\beta}^{\infty} \overline{G_3}(w)\mathrm{d}w + C_z. \end{aligned} \tag{26}$$

4.3.5. Cost Ratio Model. The cost rate in a cycle is expressed as the total cost within a cycle divided by the cycle time. Then,

$$\begin{aligned} \mathrm{TCR}(S,T) &= \frac{C(S,T)}{\mathrm{EC}(T)} = \frac{P^1C^1(S,T) + P^2C^2(S,T) + P^3C^3(S,T) + P^4C^4(S,T)}{P^1\mathrm{EC}^1(T) + P^2\mathrm{EC}^2(T) + P^3\mathrm{EC}^3(T) + P^4\mathrm{EC}^4(T)} \\ &= \frac{\begin{array}{l} P^1\left(C_h^1(S) + C_s^1(S,T) + C_m^1(T)\right) + P^2\left(C_h^2(S) + C_s^2(S,T) + C_m^2(T)\right) \\ +P^3\left(C_h^3(S) + C_s^3(S,T) + C_m^3(T)\right) + P^4\left(C_h^4(S) + C_s^4(S,T) + C_m^4(T)\right) \end{array}}{P^1\mathrm{EC}^1(T) + P^2\mathrm{EC}^2(T) + P^3\mathrm{EC}^3(T) + P^4\mathrm{EC}^4(T)}. \end{aligned} \tag{27}$$

Thus, the maintenance cost ratio model is

$$\begin{cases} \min\{\mathrm{TCR}(S,T)\}, \\ S, T \in N^*; \quad S, T > 0. \end{cases} \tag{28}$$

5. Model Solving

5.1. Calculate the Failure Rate λ_u and Maintenance Rate μ_u of Virtual Machine. Equation (1) is solved by mathematical induction:

$$\begin{aligned}\lambda_u(i) &= a\lambda_u(i-1) + (1-a)\lambda_i \\ \lambda_u(i-1) &= a\lambda_u(i-2) + (1-a)\lambda_{i-1} \\ &\vdots \\ \lambda_u(2) &= a\lambda_u(1) + (1-a)\lambda_2.\end{aligned} \tag{29}$$

$\lambda_u(1)$ is equal to λ_1.

Similarly, (2) is solved by mathematical induction:

$$\begin{aligned}t_u(i) &= ar_u(i-1) + (1-a)t_i \\ t_u(i-1) &= ar_u(i-2) + (1-a)t_{i-1} \\ &\vdots \\ t_u(2) &= ar_u(1) + (1-a)t_2.\end{aligned} \tag{30}$$

Get the average maintenance time of virtual machine $M_u(i)$, and then get its maintenance rate $\mu_u(i)$.

5.2. Solving the Optimal Maintenance Cycle and Buffer Stock. In this paper, discrete iterative algorithm is used to solve the optimal solution. The specific steps are as follows:

Step 1. To assign $S = S_{\min}$

Step 1.1. $T = T_{\min}$.

Step 1.2. To solve TCR (T, S), assign TCR (T^*, S) = TCR (T, S).

Step 1.3. $T = T + \Delta T$, to solve TCR (T, S).

Step 1.4. To judge if $T < T_{\max}$. If so, it goes to Step 1.5; otherwise, go to Step 1.6.

Step 1.5. To judge if TCR (T^*, S) > TCR (T, S). If so, assign TCR (T^*, S) = TCR (T, S), $T^* = T$, $T^* = T$; it goes to Step 1.3; otherwise, to record TCR (T^*, S), T^*, go to Step 1.5.

Step 1.6. To assign $S = S + \Delta S$, to judge if $S < S_{\max}$. If so, it goes to Step 1.1; otherwise, the program ends.

Step 2. Through Step 1, we can obtain the optimal operating cycle T^* under different stock allocation amounts S, as well as all the cost rates TCR (T^*, S). Record all the TCR (T^*, S) that we get. After sorting, it is easy to find the system's minimum average cost rateTCR $(T^*, S^*) = \min_{S_{\min} \le S \le S_{\max}}\{\text{TCR}(T^*, S)\}$ and the most joint strategy (T^*, S^*).

The flow chart of discrete iteration algorithm is shown in Figure 8.

6. Case Study

In this numerical example, the specific parameters and data of the intelligent series system were obtained from Shanghai Pangyuan Machinery Co.. The workshop has a lathe production line consisting of four machines and three buffers. By monitoring the equipment history fault record, the equipment fault parameters are summarized as follows. The original defect stage, serious defect stage, and failure stage of machine M_i are subject to exponential distribution independently. $f_x(x), f_y(y), f_z(z)$ are used to represent the probability density functions of machine deterioration in each stage, respectively. The definition of the exponential distribution function is given as follows:

$$f(x) = \lambda e^{-\lambda x}. \tag{31}$$

$\lambda_{i1}, \lambda_{i2}, \lambda_{i3}$ are used to represent the parameters in the exponential distribution of the $f_x(x), f_y(y), f_z(z)$ which are shown in Table 1.

The productivity of production line β is 30000 units per year. The buffer replenishment rate α is 6000 units per year. Shortage cost ρ = \$200 per unit. The cost of each machine monitoring process is \$800. The unit cost of corrective repair is \$15000, the unit cost of serious defect repair is \$7000, and the unit cost of original defect repair is \$4000. The maintenance time of each machine in the original defect state is supposed to be uniformly distributed between 0.5 and 1 day. The maintenance time of each machine in the serious defect state is supposed to be uniformly distributed between 2 and 5 days. The maintenance time of corrective maintenance is supposed to be uniformly distributed from 3 to 7 days. S varies from 0 to 211 units. T ranges from 0 to 105 days.

Using approximate decomposition method, the original production line is decomposed into three virtual series systems with two machines and one buffer. In the specific solution, $a = 0.2, a = 0.5, a = 0.8$ are, respectively, taken into the solution. Since the maintenance rate of each machine is the same, the maintenance rate of the decomposed virtual machine is the same as that of the original machine, so only the failure rate of the decomposed virtual machine needs to be solved.

In the case of $a = 0.2$, the failure rate $\lambda_u(i)$ of the virtual machine $M_u(i)$ solved is shown in Table 2.

In order to simplify the difficulty of solving and relate to the actual situation, only the case where period T and buffer stock S are integers is considered in this paper. The cost ratio model is a double integer parameter nonlinear programming problem. One discrete iteration algorithm is used to solve the model. The optimal monitoring time T_1, T_2, T_3 of the machines $M_u(1), M_u(2), M_u(3)$ is 29, 27, 27 days. The optimal stock allocation amounts S_1, S_2, S_3 of buffers B_1, B_2, B_3 are 79, 80, 79 units. Figure 9 shows the change of the cost rate of machines $M_u(1)$, $M_u(2)$, $M_u(3)$ with S, T.

When $a = 0.2$, the operating cycle of each machine in the intelligent series system, the buffer stock allocation amount, and the corresponding lowest cost rate are shown in Table 3.

Similarly, in the case of $a = 0.5$, the failure rate $\lambda_u(i)$ of the virtual machine $M_u(i)$ solved is shown in Table 4.When $a = 0.5$, the operating cycle of each machine in the production line, the buffer stock allocation amount, and the corresponding lowest cost rate are shown in Table 5.

Similarly, in the case of $a = 0.5$, the failure rate $\lambda_u(i)$ of the virtual machine $M_u(i)$ solved is shown in Table 6.When $a = 0.8$, the monitoring time of each machine in the intelligent series system, the buffer stock allocation amount, and the corresponding lowest cost rate are shown in Table 7.

6.1. Result Analysis. For different influence factors a, the monitoring time and buffer stock allocation are obtained. Table 8 is a comparison of the optimal monitoring time for each machine under different a. Table 9 is a comparison of the best stocks for each buffer under different a.

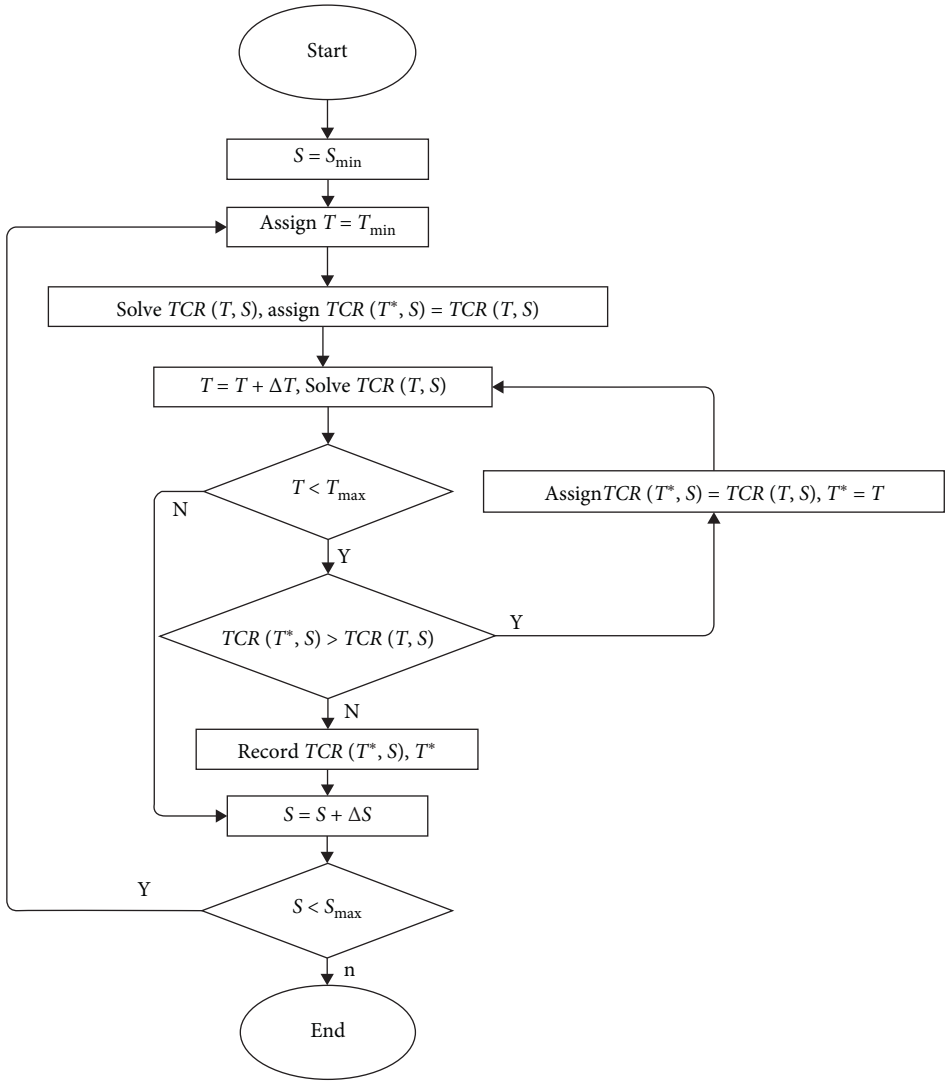


Figure 8: Flow chart of discrete iteration algorithm.

Table 1: Related parameters of failure rate distribution, unit: 1/year.

	M_1	M_2	M_3
λ_{i1}	$\lambda_{11} = 1.0$	$\lambda_{21} = 1.4$	$\lambda_{31} = 1.2$
λ_{i2}	$\lambda_{12} = 1.2$	$\lambda_{22} = 1.6$	$\lambda_{32} = 1.5$
λ_{i3}	$\lambda_{13} = 1.5$	$\lambda_{23} = 1.8$	$\lambda_{33} = 1.8$

Table 2: Failure rate of virtual machine $M_u(i)$ for $a = 0.2$, unit: 1/year.

$a = 0.2$	$M_u(1)$	$M_u(2)$	$M_u(3)$
$\lambda_u(i1)$	$\lambda_u(11) = 1.0$	$\lambda_u(21) = 1.32$	$\lambda_u(31) = 1.224$
$\lambda_u(i2)$	$\lambda_u(12) = 1.2$	$\lambda_u(22) = 1.52$	$\lambda_u(32) = 1.504$
$\lambda_u(i3)$	$\lambda_u(13) = 1.5$	$\lambda_u(23) = 1.74$	$\lambda_u(33) = 1.788$

As can be seen from Table 8, with the increase of a, the operational cycle of the same machine gradually increases, and the inventory of the same buffer gradually decreases. Considering the actual situation, the smaller the a is, the higher the importance of machine M_i will be. Therefore, the shorter the operation cycle is, the shorter the monitoring time is, the higher the maintenance frequency is, and the higher the inventory allocated by the corresponding buffer will be. Therefore, enterprises can choose the value of impact factor a according to the importance of the machine in the production line, so as to obtain more accurate preventive maintenance strategy and buffer stock allocation strategy.

$a = 0.2$ is fixed. For machine M_1, the optimal inventory and minimum cost rate under different monitoring time T and the optimal monitoring time and minimum cost rate under different inventory S were obtained by solving the problem, as shown in Table 9. As can be seen from Table 9, the increase or decrease of T and the increase or decrease of S will lead to the increase of the cost rate. If T is too small, the number of monitoring and maintenance processes will increase, which will lead to the increase of maintenance cost and the frequent shutdown of the machine. On the contrary, if T is too large, the possibility of machine failure shutdown will be greater, and the shortage cost will also increase. If S is

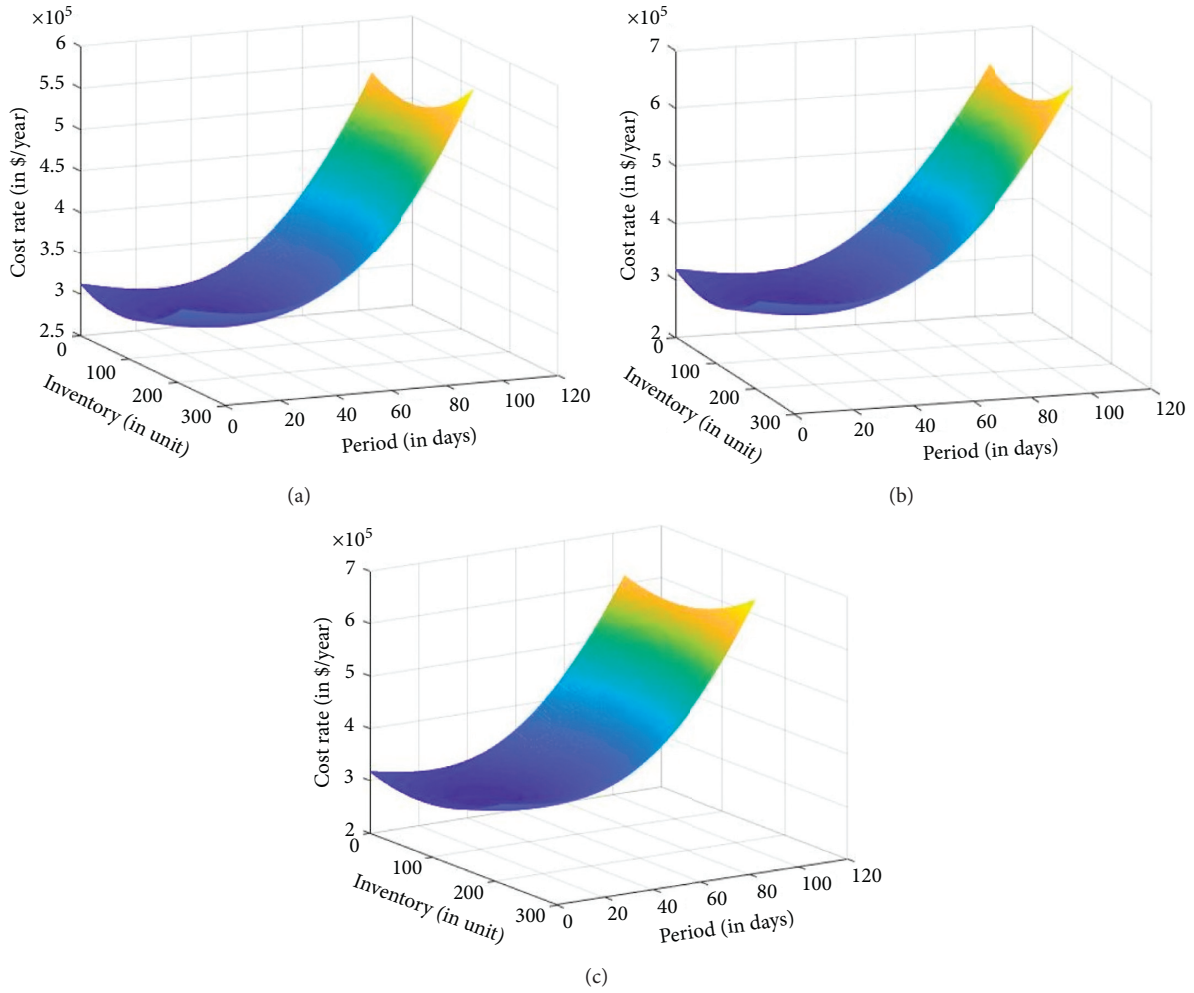


Figure 9: Diagram of the cost rate (in $/year) of machines $M_u(1)$, $M_u(2)$, $M_u(3)$ with the monitoring time T(in days) and buffer stock S (in units).

Table 3: The lowest cost rate with the optimal monitoring time of the machine of the intelligent series system and the optimal buffer stock for $a = 0.2$

$a = 0.2$	$i = 1$	$i = 2$	$i = 3$
T_i of machine M_i (in days)	29	27	27
S_i of buffer B_i (in units)	79	80	79
Cost rateTCR (S, T)(in $/year)	282137.8	280065.2	281066.7

Table 4: Failure rate of virtual machine$M_u(i)$ for $a = 0.5$, unit: 1/year.

$a = 0.5$	$M_u(1)$	$M_u(2)$	$M_u(3)$
$\lambda_u(i1)$	$\lambda_u(11) = 1.0$	$\lambda_u(21) = 1.2$	$\lambda_u(31) = 1.2$
$\lambda_u(i2)$	$\lambda_u(12) = 1.2$	$\lambda_u(22) = 1.4$	$\lambda_u(32) = 1.45$
$\lambda_u(i3)$	$\lambda_u(13) = 1.5$	$\lambda_u(23) = 1.65$	$\lambda_u(33) = 1.725$

Table 5: The lowest cost rate with the optimal monitoring time of the machine of the intelligent series system and the optimal buffer stock for $a = 0.5$.

$a = 0.5$	$i = 1$	$i = 2$	$i = 3$
T_i of machine M_i (in days)	29	28	27
S_i of buffer B_i (in units)	79	79	79
Cost rateTCR (S, T)(in $/year)	282137.8	280804.5	281065.1

Table 6: Failure rate of virtual machine $M_u(i)$ for $a = 0.8$, unit: 1/year.

$a = 0.8$	$M_u(1)$	$M_u(2)$	$M_u(3)$
$\lambda_u(i1)$	$\lambda_u(11) = 1.0$	$\lambda_u(21) = 1.08$	$\lambda_u(31) = 1.104$
$\lambda_u(i2)$	$\lambda_u(12) = 1.2$	$\lambda_u(22) = 1.28$	$\lambda_u(32) = 1.324$
$\lambda_u(i3)$	$\lambda_u(13) = 1.5$	$\lambda_u(23) = 1.56$	$\lambda_u(33) = 1.608$

Table 7: The lowest cost rate with the optimal monitoring time of the machine of the intelligent series system and the optimal buffer stock with different a.

$a = 0.8$	$i = 1$	$i = 2$	$i = 3$
T_i of machine M_i (in days)	29	28	28
S_i of buffer B_i (in units)	79	79	79
Cost rateTCR (S, T)(in \$/year)	282137.8	281588.8	281531.0

Table 8: The optimal monitoring time T and the optimal stock S of each machine in the intelligent series system with different a.

	$a = 0.2$	$a = 0.5$	$a = 0.8$
T_1/S_1 of machine M_1 (in days/units)	29/79	29/79	29/79
T_2/S_2 of machine M_2 (in days/units)	27/80	28/79	28/79
T_3/S_3 of machine M_3 (in days/units)	27/79	27/79	28/79

Table 9: The optimal inventory S and the minimum cost ratio under different T and the optimal monitoring time T and the minimum cost rate under different S.

Machine M_1					Machine M_1
T	S^*	TCR* (S, T)	S	T^*	TCR* (S, T)
9	78	290802.0	19	29	291379.6
19	78	284631.7	39	29	286215.1
29	79	282137.8	59	29	283134.5
39	79	285132.4	79	29	283137.8
49	79	295151.5	99	29	283225.1
59	78	313425.6	119	29	286396.3
69	77	340867.0	139	29	291651.4
79	76	778048.1	159	28	298989.5
89	75	425204.7	179	28	308394.0
99	74	482245.5	199	28	319878.2

too small, it will be more likely to be out of stock, which will lead to the increase of shortage cost. If S is too large, it will inevitably lead to an increase in inventory cost.

From a practical point of view, the model results are consistent with the reality. If the monitoring time is 9 days, this means that if the detection is carried out every 9 days, the maintenance cost will be too high. On the other hand, if the buffer stock is replenished to 200 pieces, the inventory cost is high. The cost rates are highest in these extremes.

6.2. Result Comparison. The maintenance cost rate model established in this paper is combined with the three-stage time delay theory. According to the concept of three-stage fault process, the states of the system include normal, original defect, serious defect, and fault state. Compared with the traditional two-stage time delay theory, if the machine failure can be detected in the original defect state, not only the money cost but also the time cost can be saved. In this section, $a = 0.2$ is fixed. For machine M_1, the maintenance strategy proposed in this paper is compared with the maintenance strategy without buffer stock and the maintenance strategy based on two-stage time delay.

6.2.1. Comparison with a Maintenance Strategy without Buffer Stock. Buffer was added to the maintenance system in this paper. In order to illustrate the effectiveness of the model, and Table 10 compares the optimal monitoring time and the minimum cost rate of machines M_1, M_2, M_3 with and without buffer stock in the case of $a = 0.2, a = 0.5, a = 0.8$. Not taking buffer stock into account means that the buffer stock is 0. Table 10 shows that the cost ratio is smaller when buffer stock is taken into account than when buffer stock is not taken into account. It shows that the preventive maintenance strategy considering buffer stock is optimal, feasible, and effective.

6.2.2. Comparison with the Maintenance Strategy Based on the Two-Stage Time Delay Theory. According to the traditional time delay theory, there are three states of a machine: normal, defect, and failure. The defect state and fault state

Table 10: Comparison of two maintenance strategies for $a = 0.2$, $a = 0.5$, and $a = 0.8$.

$a = 0.2$	S_1	T_1^*	TCR* (S,T)
Machine M_1	79	29	282137.8
	0	28	298210.2
Machine M_2	80	27	280065.2
	0	26	302058.2
Machine M_3	79	27	281066.7
	0	26	301232.9
$a = 0.5$	S_1	T_1^*	TCR* (S,T)
Machine M_1	79	29	282137.8
	0	28	298210.2
Machine M_2	79	28	280804.5
	0	27	300807.4
Machine M_3	79	27	281066.7
	0	27	300559.5
$a = 0.8$	S_1	T_1^*	TCR* (S,T)
Machine M_1	79	29	282137.8
	0	28	298210.2
Machine M_2	79	28	281588.8
	0	28	299125.3
Machine M_3	79	28	281531.0
	0	27	299209.3

occurred in T_y and T_z. There are three different situations for adopting preventive maintenance based on monitoring status, machine monitoring time T, and buffer stock S.

(1) $0 < T < T_y$. The state monitoring time of the machine occurs before the defect time. Machine status is in a non-defective state, and it is unnecessary to execute any maintenance action. After that, in order to prevent the failure to detect the defect in time, it is necessary to execute a state monitoring process on the machine every day until the original defect is detected. Buffer stock level change during a running cycle is shown in Figure 10. Preventive maintenance of machine is executed under the defect state.

The probability of Twithin $[10, T_y]$:

$$P\left(0 < T < T_y\right) = P_1 = \int_T^{\infty} f_y(y)\mathrm{d}y. \tag{32}$$

Operation cycle of machine for $0 < T < T_y$:

$$\mathrm{EC}_1(T) = E\left(W_2\right) + E\left(T_y\right). \tag{33}$$

Inventory holding cost in a cycle:

$$C_{h1}(S) = h\left[\frac{S^2}{2}\left(\frac{\alpha+\beta}{\alpha\beta}\right) + (y-T)S\right]. \tag{34}$$

Shortage cost in a cycle:

$$C_{s1}(S,T) = \rho\beta\int_{S/\beta}^{\infty}\overline{G_2}(w)\mathrm{d}w. \tag{35}$$

Maintenance cost in a cycle:

$$C_{m1}(T) = C_y + (y - T + 1)C_r. \tag{36}$$

The expected cost in a cycle for $0 < T < T_y$ is the sum of inventory holding cost, shortage cost, and maintenance cost. The expected cost $C_1(S,T)$ is obtained as

$$\begin{aligned} C_1(S,T) &= C_{h1}(S) + C_{s1}(S,T) + C_{m1}(T) \\ &= h\left[\frac{S^2}{2}\left(\frac{\alpha+\beta}{\alpha\beta}\right) + (y-T)S\right] + \rho\beta\int_{S/\beta}^{\infty}\overline{G_2}(w)\mathrm{d}w + C_y + (y-T+1)C_r. \end{aligned} \tag{37}$$

(2) $Ty < T < T_z$. The state monitoring time of the machine occurs after the defect time and before the breakdown time. Machine status is in a defect operation state, and it is necessary to execute preventive maintenance of the defect state. After the buffer stock is replenished, the machine status is monitored immediately. Buffer stock level change during a running cycle is shown in Figure 11.

The probability of T within $[T_y, T_z]$:

$$P\left(T_y < T < T_z\right) = P_2 = \int_0^T\int_{T-y}^{\infty} f_y(y)f_z(z)\mathrm{d}x\mathrm{d}y = \int_0^T f_y(y)\left(1 - F_z(T-y)\right)\mathrm{d}y. \tag{38}$$

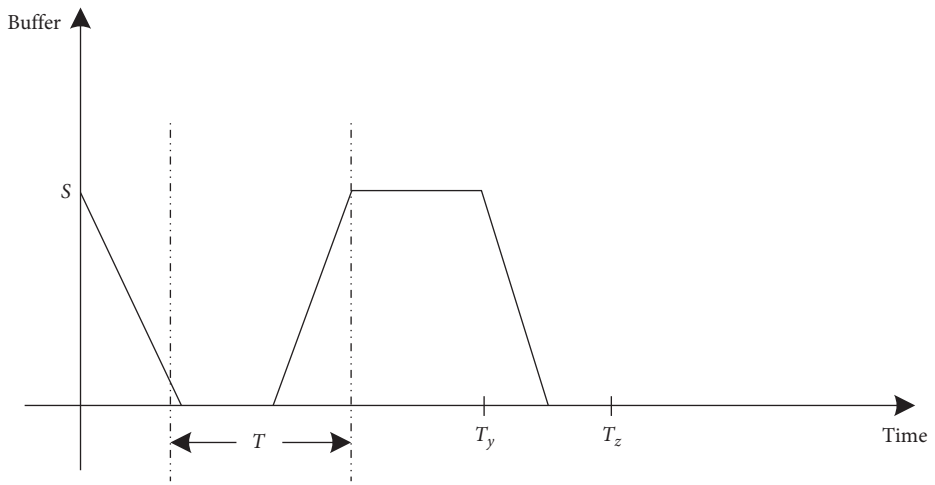


Figure 10: Buffer stock change diagram in a cycle for $0 < T < T_y$.

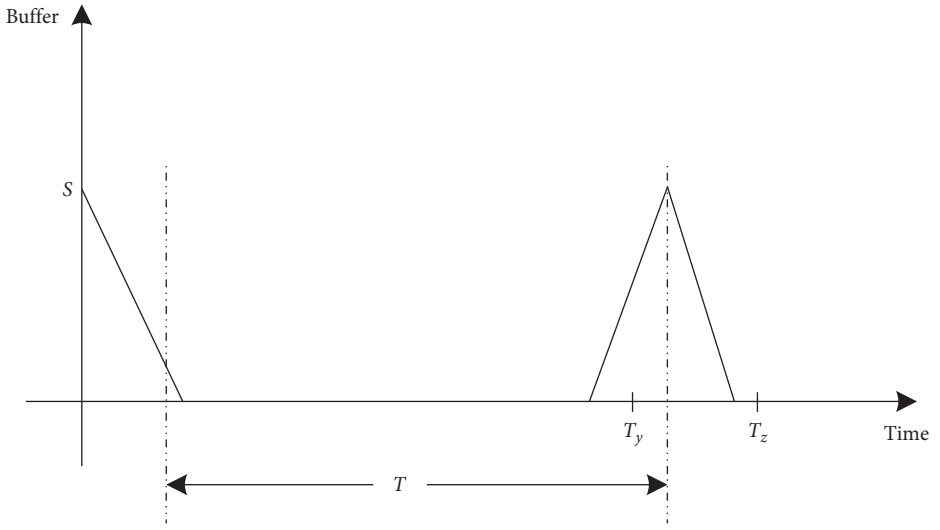


Figure 11: Buffer stock change diagram in a cycle for $T_y < T < T_z$.

Operating cycle of the machine for $T_y < T < T_z$:

$$\mathrm{EC}_2(T) = E(W_2) + T. \tag{39}$$

Inventory holding cost in a cycle:

$$C_{h2}(S) = h\left[\frac{S^2}{2}\left(\frac{\alpha+\beta}{\alpha\beta}\right)\right]. \tag{40}$$

Shortage cost in a cycle:

$$C_{s2}(S,T) = \rho\beta\int_{S/\beta}^{\infty}\overline{G_2}(w)\mathrm{d}w. \tag{41}$$

Maintenance cost in a cycle:

$$C_{m2}(T) = C_y + C_r. \tag{42}$$

The expected cost in a cycle for $T_y < T < T_z$ is the sum of inventory holding cost, shortage cost, and maintenance cost. The expected cost $C_2(S,T)$ is obtained as

$$C_2(S,T) = C_{h2}(S) + C_{s2}(S,T) + C_{m2}(T) = h\left[\frac{S^2}{2}\left(\frac{\alpha+\beta}{\alpha\beta}\right)\right] + \rho\beta\int_{S/\beta}^{\infty}\overline{G_2}(w)\mathrm{d}w + C_y + C_r. \tag{43}$$

(3) $T > T_z$. In this case, the machine fails before the state detection is carried out. The planned state detection takes place after the failure of the machine. At this time, the corrective maintenance action is executed. The planned state detection takes place after the failure of the machine. Buffer stock has not been replenished, or the production of buffer stock has not started before the failure shutdown. Thus, the buffer stock change during a running cycle is shown in Figure 12.

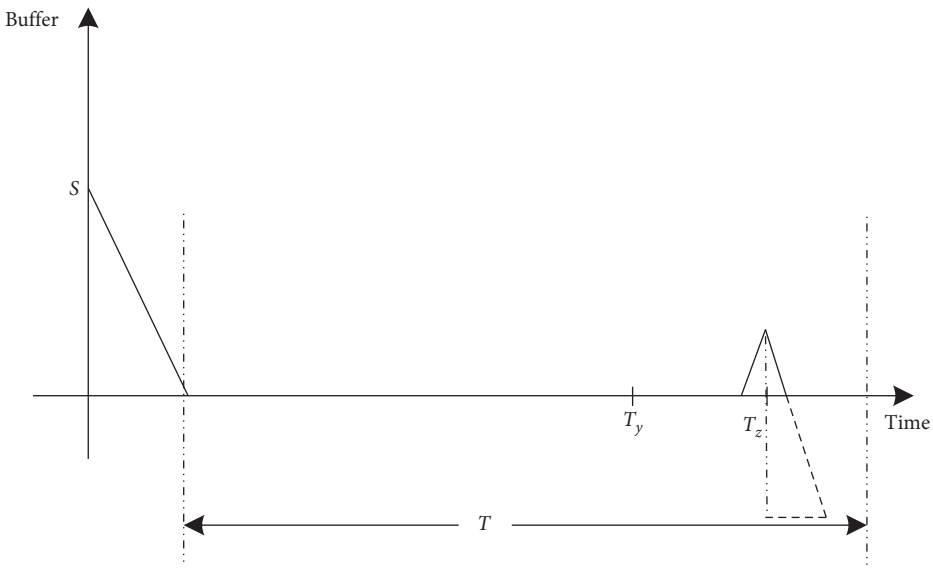


Figure 12: Buffer stock change diagram in a cycle for $T > T_z$.

The probability of T within $[T_z, \infty]$:

$$P(T > T_z) = P_3 = \int_0^T \int_0^{T-Y} f_y(y) f_z(z) \mathrm{d}y \mathrm{d}z = \int_0^T f_y(y) F_z(T-y) \mathrm{d}y. \tag{44}$$

Operating cycle of the machine for $T > T_z$:

$$\mathrm{EC}_3(T) = E(W_3) + E(T_z). \tag{45}$$

Inventory holding cost in a cycle:

$$C_{h3}(S) = \frac{h}{2}\left[S - \alpha\left(T - E(T_z)\right)\right]\left[\frac{(\alpha+\beta)S}{\alpha\beta} - \left(\frac{\alpha+\beta}{\beta}\right)\left(T - E(T_z)\right)\right]. \tag{46}$$

Shortage cost in a cycle:

$$C_{s3}(S,T) = \rho\beta \int_{(S-\alpha(T-E(T_z)))/\beta}^{\infty} \overline{G_3}(w) \mathrm{d}w. \tag{47}$$

Maintenance cost in a cycle:

$$C_{m3}(T) = C_z. \tag{48}$$

The expected cost in a cycle for $T > T_z$ is the sum of inventory holding cost, shortage cost, and maintenance cost. The expected cost $C_3(S,T)$ is obtained as

$$\begin{aligned} C_3(S,T) &= C_{h3}(S) + C_{s3}(S,T) + C_{m3}(T) \\ &= \frac{h}{2}\left[S - \alpha\left(T - E(T_z)\right)\right]\left[\frac{(\alpha+\beta)S}{\alpha\beta} - \left(\frac{\alpha+\beta}{\beta}\right)\left(T - E(T_z)\right)\right] \\ &\quad + \rho\beta \int_{S-\alpha(T-E(T_z))/\beta}^{\infty} \overline{G_3}(w) \mathrm{d}w + C_z. \end{aligned} \tag{49}$$

The cost rate in a cycle is expressed as follows:

$$\mathrm{TCR}(S,T) = \frac{C(S,T)}{\mathrm{EC}(T)} = \frac{P_1 C_1(S,T) + P_2 C_2(S,T) + P_3 C_3(S,T)}{P_1 \mathrm{EC}_1(T) + P_2 \mathrm{EC}_2(T) + P_3 \mathrm{EC}_3(T)}. \tag{50}$$

Based on the traditional two-stage time delay theory, the machine preventive maintenance model considering buffer stock is established as follows:

$$\begin{cases} \min\{\mathrm{TCR}(S,T),\} \\ S, T \in N^*; \quad S, T > 0. \end{cases} \tag{51}$$

The $a = 0.2$ is fixed. For M_1, all parameters in the solution remained unchanged, and the model was solved. The optimal monitoring time of machine M_1 is 3 days, the optimal stock of buffer is 211 units, and the minimum maintenance cost of machine in one year is \$318378.7, as shown on the left of Figure 12. Considering that the buffer stock has reached the upper limit previously given, the value of S is adjusted, and $0 < S < 400$ is set to solve the problem. The optimal monitoring time T of machine M_1 is 3 days, the optimal stock S of buffer is 387 units, and the minimum maintenance cost of machine in one year is \$304570.2, as shown on the right of Figure 13. As can be seen from the results, based on the traditional two-stage time delay model, the monitoring time is short and the buffer stock is high. This is because there is no distinction between the original defects and the serious defects of the machine, and the machine status cannot be accurately detected. In order to prevent the machine from being shut down, it is necessary to carry out regular monitoring, which is consistent with the actual situation. The maintenance strategy based on three-stage time delay theory can save \$22432.4 per year compared with the maintenance strategy based on two-stage time delay theory. The specific comparison is shown in Table 11.

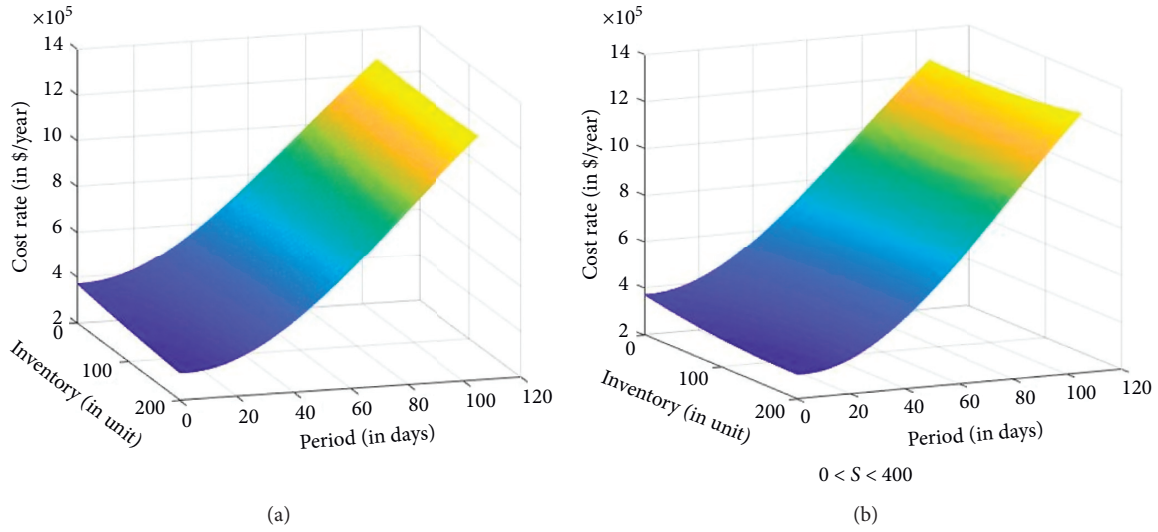


Figure 13: Diagram of the cost rate (in $/year) of machine M_1 with T (in days) and S (in units).

Table 11: Comparison of two maintenance strategies.

M_1	S_1	T_1^*	TCR* (S,T)
Maintenance strategy based on three-stage time delay theory	79	29	282137.8
Maintenance strategy based on two-stage time delay theory	387	3	304570.2
Maintenance strategy based on two-stage time delay theory	79	29	445867.4

7. Conclusion

In this paper, a new method is proposed to solve the preventive maintenance problem of intelligent series system with buffer stock. For the intelligent series system with inventory buffer, the series system is decomposed into several virtual series systems with two machines and one buffer by approximate decomposition method. The failure rate and maintenance rate of the decomposed virtual machine are calculated by mathematical induction. The influence factor is introduced here, and the enterprise can determine the value of the influence factor according to the importance of different machines in the series system. For each virtual series system, a preventive maintenance model was built with the lowest cost rate as the objective function and the monitoring time and buffer stock as independent variables. The preventive maintenance model is combined with the three-stage time delay theory to better simulate the equipment degradation process. Finally, a case is used to verify the validity of the model.

The maintenance strategy in this paper is compared with the maintenance strategy without buffer stock and the maintenance strategy based on the two-stage time delay. It is proved that the proposed maintenance strategy based on the three-stage time delay theory is optimal. Taking the impact factor $a = 0.2$ as an example, compared with no buffer stock maintenance strategy, the annual maintenance cost of machine M_1 can save \$16072.4. Compared with the traditional two-stage time delay maintenance strategy, the annual maintenance cost of machine M_1 can save \$22432.4. Therefore, the maintenance strategy proposed in this paper can be well used in the maintenance of the series system, which can save a lot of money for enterprises

The value of impact factor a should be determined according to the importance of the machine on the production line. On the basis of the research in this study, the effective method can be adopted in following research, and the most accurate influence factor can be obtained for different machines in the series system.

Data Availability

The underlying data supporting the results of our study can be found in the article, including, where applicable, hyperlinks to publicly archived datasets analyzed or generated during the study

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work presented in this paper has been supported by Grants from National Natural Science Foundation of China (Nos. 71632008, 71840003, and 51875359), Natural Science Foundation of Shanghai (Nos. 19ZR1435600 and 20ZR1428600), and Humanity and Social Science Planning Foundation of Ministry of Education of China (No. 20YJAZH068).

References

[1] F. Wu, S. A. Niknam, and J. E. Kobza, "A cost effective degradation-based maintenance strategy under imperfect

repair," *Reliability Engineering & System Safety*, vol. 144, pp. 234–243, 2015.

[2] G. Q. Cheng, B. H. Zhou, and L. Li, "Integrated production, quality control and condition-based maintenance for imperfect production systems," *Reliability Engineering & System Safety*, vol. 175, pp. 251–264, 2018.

[3] R. Rooeinfar, S. Raissi, and V. Ghezavati, "Stochastic flexible flow shop scheduling problem with limited buffers and fixed interval preventive maintenance: a hybrid approach of simulation and metaheuristic algorithms," *Simulation*, vol. 95, no. 6, pp. 509–528, 2019.

[4] Z. Zhang, Q. Tang, and C. Lu, "Multi-objective U-shaped assembly line balancing under machine deterioration and preventive maintenance," *ICIC Express Letters*, vol. 12, no. 11, pp. 1123–1130, 2018.

[5] K. Kang and V. Subramaniam, "Joint control of dynamic maintenance and production in a failure-prone manufacturing system subjected to deterioration," *Computers & Industrial Engineering*, vol. 119, pp. 309–320, 2018.

[6] B. Bouslah, A. Gharbi, and R. Pellerin, "Joint production, quality and maintenance control of a two-machine line subject to operation-dependent and quality-dependent failures," *International Journal of Production Economics*, vol. 195, pp. 210–226, 2018.

[7] M. Motlagh, P. Azimi, M. Amiri, and G. Madraki, "An efficient simulation optimization methodology to solve a multi-objective problem in unreliable unbalanced production lines," *Expert Systems with Applications*, vol. 138, Article ID 112836, 2019.

[8] L. Wang, Z. Lu, and Y. Ren, "Joint production control and maintenance policy for a serial system with quality deterioration and stochastic demand," *Reliability Engineering & System Safety*, vol. 199, Article ID 106918, 2020.

[9] X. Wang, S. Guo, J. Shen, and Y. Liu, "Optimization of preventive maintenance for series manufacturing system by differential evolution algorithm," *Journal of Intelligent Manufacturing*, vol. 31, no. 3, pp. 745–757, 2020.

[10] S. Ozcan and F. Simsir, "A new model based on artificial bee colony algorithm for preventive maintenance with replacement scheduling in continuous production lines," *Engineering Science and Technology, an International Journal*, vol. 22, no. 6, pp. 1175–1186, 2019.

[11] B. Zhou, Y. Qi, and Y. Liu, "Proactive preventive maintenance policy for buffered serial production systems based on energy saving opportunistic windows," *Journal of Cleaner Production*, vol. 253, 2020.

[12] J. Wijngaard, "The effect of interstage buffer storage on the output of two unreliable production units in series, with different production rates," *AIIE Transactions*, vol. 11, no. 1, pp. 42–47, 1979.

[13] R. I. Zequeira, B. Prida, and J. E. Valdés, "Optimal buffer inventory and preventive maintenance for an imperfect production process," *International Journal of Production Research*, vol. 42, no. 5, pp. 959–974, 2004.

[14] R. I. Zequeira, J. E. Valdes, and C. Berenguer, "Optimal buffer inventory and opportunistic preventive maintenance under random production capacity availability," *International Journal of Production Economics*, vol. 111, no. 2, pp. 686–696, 2008.

[15] A. Mohtashami, "A new hybrid method for buffer sizing and machine allocation in unreliable production and assembly lines with general distribution time-dependent parameters," *The International Journal of Advanced Manufacturing Technology*, vol. 74, no. 9–12, pp. 1577–1593, 2014.

[16] L. Demir, S. Tunali, and A. Løkketangen, "A tabu search approach for buffer allocation in production lines with unreliable machines," *Engineering Optimization*, vol. 43, no. 2, pp. 213–231, 2011.

[17] B. K. Sett, S. Sarkar, and B. Sarkar, "Optimal buffer inventory and inspection errors in an imperfect production system with preventive maintenance," *The International Journal of Advanced Manufacturing Technology*, vol. 90, no. 1–4, pp. 545–560, 2017.

[18] M.-C. Fitouhi, M. Nourelfath, and S. B. Gershwin, "Performance evaluation of a two-machine line with a finite buffer and condition-based maintenance," *Reliability Engineering & System Safety*, vol. 166, pp. 61–72, 2017.

[19] G. P. Kiesmüller and F. E. Sachs, "Spare parts or buffer? how to design a transfer line with unreliable machines," *European Journal of Operational Research*, vol. 284, no. 1, pp. 121–134, 2020.

[20] N. Nahas, "Buffer allocation and preventive maintenance optimization in unreliable production lines," *Journal of Intelligent Manufacturing*, vol. 28, no. 1, pp. 85–93, 2017.

[21] M. Zandieh, M. N. Joreir-Ahmadi, and A. Fadaei-Rafsanjani, "Buffer allocation problem and preventive maintenance planning in non-homogenous unreliable production lines," *The International Journal of Advanced Manufacturing Technology*, vol. 91, no. 5–8, pp. 2581–2593, 2017.

[22] R. Lopes, "Integrated model of quality inspection, preventive maintenance and buffer stock in an imperfect production system," *Computers & Industrial Engineering*, vol. 126, pp. 650–656, 2018.

[23] Y. Kang and F. Ju, "Flexible preventative maintenance for serial production lines with multi-stage degrading machines and finite buffers," *IISE Transactions*, vol. 51, no. 7, pp. 777–791, 2019.

[24] A. Alfieri, A. Matta, and E. Pastore, "The time buffer approximated buffer allocation problem: a row-column generation approach," *Computers & Operations Research*, vol. 115, Article ID 104835, 2020.

[25] Y. Dallery, R. David, and X.-L. Xie, "Approximate analysis of transfer lines with unreliable machines and finite buffers," *IEEE Transactions on Automatic Control*, vol. 34, no. 9, pp. 943–953, 1989.

[26] K.-C. Jeong and Y.-D. Kim, "Performance analysis of assembly/disassembly systems with unreliable machines and random processing times," *IIE Transactions*, vol. 30, no. 1, pp. 41–53, 1998.

[27] K. Dhouib, A. Gharbi, and N. Landolsi, "Throughput assessment of mixed-model flexible transfer lines with unreliable machines," *International Journal of Production Economics*, vol. 122, no. 2, pp. 619–627, 2009.

[28] L. Li, Y. Qian, Y. Yang, and K. Du, "A common model for the approximate analysis of tandem queueing systems with blocking," *IEEE Transactions on Automatic Control*, vol. 61, no. 7, pp. 1780–1793, 2016.

[29] S. Xi, Q. Chen, J. MacGregor Smith, N. Mao, A. Yu, and H. Zhang, "A new method for solving buffer allocation problem in large unbalanced production lines," *International Journal of Production Research*, vol. 58, no. 22, pp. 6846–6867, 2020.

[30] N. Nahas and M. Nourelfath, "Joint optimization of maintenance, buffers and machines in manufacturing lines," *Engineering Optimization*, vol. 50, no. 1, pp. 37–54, 2018.

[31] B. Xia, C. Wang, Y. Gao, Y. Peng, and L. Liu, "A new approach to the analysis of homogeneous transfer lines with unreliable buffers subject to time-dependent failure," *International*

Journal of Production Research, vol. 58, no. 21, pp. 6707–6723, 2020.

[32] Y. Bai, J. Tu, M. Yang, L. Zhang, and P. Denno, "A new aggregation algorithm for performance metric calculation in serial production lines with exponential machines: design, accuracy and robustness," *International Journal of Production Research*, pp. 1–18, 2020.

[33] A. H. Christer and W. M. Waller, "Delay time models of industrial inspection maintenance problems," *Journal of the Operational Research Society*, vol. 35, no. 5, pp. 401–406, 1984.

[34] W. Wang, "Models of inspection, routine service, and replacement for a serviceable one-component system," *Reliability Engineering & System Safety*, vol. 116, pp. 57–63, 2013.

[35] J. Zhao, A. H. C. Chan, C. Roberts, and K. B. Madelin, "Reliability evaluation and optimisation of imperfect inspections for a component with multi-defects," *Reliability Engineering & System Safety*, vol. 92, no. 1, pp. 65–73, 2007.

[36] J. Gomes da Silva and R. S. Lopes, "An integrated framework for mode failure analysis, delay time model and multi-criteria decision-making for determination of inspection intervals in complex systems," *Journal of Loss Prevention in the Process Industries*, vol. 51, pp. 17–28, 2018.

[37] M. Mahmoudi, A. Elwany, K. Shahanaghi, and M. R. Gholamian, "A delay time model with multiple defect types and multiple inspection methods," *IEEE Transactions on Reliability*, vol. 66, no. 4, pp. 1073–1084, 2017.

[38] W. Wang, F. Zhao, and R. Peng, "A preventive maintenance model with a two-level inspection policy based on a three-stage failure process," *Reliability Engineering & System Safety*, vol. 121, pp. 207–220, 2014.

Hindawi
Shock and Vibration
Volume 2021, Article ID 6675078, 11 pages
https://doi.org/10.1155/2021/6675078

Research Article

A New Support Vector Regression Model for Equipment Health Diagnosis with Small Sample Data Missing and Its Application

Qinming Liu,[1] **Wenyi Liu**,[1] **Jiajian Mei**,[1] **Guojin Si**,[2] **Tangbin Xia**,[2] **and Jiarui Quan**[1]

[1]*Department of Industrial Engineering, Business School, University of Shanghai for Science and Technology, 516 Jungong Road, Shanghai 200093, China*
[2]*State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, SJTU-Fraunhofer Center, Shanghai 200240, China*

Correspondence should be addressed to Qinming Liu; lqm0531@163.com

Received 4 December 2020; Revised 7 January 2021; Accepted 8 February 2021; Published 25 February 2021

Academic Editor: Gerardo Silva-Navarro

Actually, it is difficult to obtain a large number of sample data due to equipment failure, and small sample data may also be missing. This paper proposes a novel small sample data missing filling method based on support vector regression (SVR) and genetic algorithm (GA) to improve equipment health diagnosis effect. First, the genetic algorithm is used to optimize support vector regression, and a new method GA-SVR can be proposed. The GA-SVR model is trained by using other data of the variable to which the missing data belongs, and the single-variable prediction method can be obtained. The correlation analysis is used to reconstruct the training set, and the GA-SVR is trained by using the data of the variables related to the missing data to obtain the multivariate prediction method. Then, the dynamic weight is presented to combine the single-variable prediction method with the multiple-variable prediction method based on certain principles, and the missing data are filled with the combined prediction methods. The filled data are used as input of GA-SVM to diagnose equipment failure. Finally, a case study is given to verify the applicability and effectiveness of the proposed method.

1. Introduction

For equipment health diagnosis, complete monitoring data is the premise and foundation for an accurate diagnosis. However, in the actual engineering application, many monitoring sample data are incomplete, including small sample, unbalanced sample, and sample data missing. In the collection of sample data, equipment may not be able to operate normally due to fault, or it can be affected by the environment, and the effective monitoring data collected is less, resulting in less failure sample data. The sample data may also be missing due to abnormal data transmission, sensor repair and replacement, or human factors. This paper importantly considers the condition of small sample data missing.

Recently, with the rapid development of technology, equipment health diagnosis has been widely concerned by a large number of experts and scholars. The intelligent diagnosis methods applied to equipment health diagnosis mainly include expert system (ES), neural networks (NNs), and support vector machine (SVM).

For the expert system, Husain [1] expanded the fault diagnosis of the power transformer, proposed a fuzzy logic expert system for early fault diagnosis of the transformer, and improved the shortcomings of traditional transformer fault diagnosis methods. Berredjem and Benidir [2] proposed a fuzzy expert system based on an improved range overlap method and similarity division method to solve the problem of high noise in bearing fault data. The system was used to realize accurate bearing fault diagnosis, and the feasibility of the model was verified by an example analysis. Cheriet et al. [3] proposed an expert system based on fuzzy logic, which used stator current signal pair for fault diagnosis, and verified the feasibility of the expert system for fault diagnosis of doubly fed wind turbines through simulation experiments. Xu et al. [4] carried out a series of

researches on the fault diagnosis of marine diesel engines, proposed a diagnosis expert system based on belief rules, and applied the proposed method to the abnormal wear detection of marine diesel engines, indicating that the method had good accuracy and stability. Equipment health diagnosis method based on the expert system can acquire knowledge from diagnosis examples, but this method does not have the ability to automatically acquire new knowledge, and the fault tolerance is relatively poor. Thus, the fault diagnosis method based on the expert system has great limitations in practical application.

For neural networks, Xing et al. [5] constructed an automatic fault diagnosis method for reciprocating compressors based on information entropy and radial basis function neural networks. The test results showed that the fault diagnosis method can effectively improve the accuracy of automatic fault diagnosis and the practicability of the condition monitoring system. Yang et al. [6] analyzed the fault diagnosis of rotating machinery, proposed an intelligent diagnosis method based on long-term and short-term memory recurrent neural network, and detected and classified the fault with the help of the correlation between time and space. Gunerkar et al. [7] established a rolling bearing fault diagnosis model based on an artificial neural network (ANN) and applied wavelet transform to preprocess the original signal to extract fault features. ANN and the k-nearest neighbor were used for fault classification of rolling bearing, and the validity of the model was verified by test. In order to solve the problem of end-to-end fault diagnosis of rotating machinery, Wu et al. [8] constructed a one-dimensional CNN model which can directly learn features from the original signal, applied it to the fault diagnosis of the fixed gearbox and planetary gearbox, and showed that the model had high diagnostic accuracy. Han et al. [9] proposed a method for fault diagnosis of the planetary gearbox by using an expanded neural network, which expanded the receiving domain by two times, so as to enhance the learning ability of fault features and improve the diagnosis accuracy. The fault diagnosis method based on an artificial neural network often needs a large number of fault samples to train the neural network, but it is difficult to obtain enough fault data in practical engineering applications. In addition, the neural network has the disadvantages of slow convergence, overfitting, and ease to fall into the local optimal value, which will have a negative impact on the diagnostic accuracy of the equipment.

For the support vector machine, Huang and Fei et al. [10, 11] used the SVM model for equipment fault diagnosis and verified that the model has high accuracy and good generalization ability. Yang et al. [12] established an SVM fault classification model using an ant colony algorithm and verified the effectiveness of the model. Zhang et al. [13] combined SVM with an improved imperialist competitive algorithm and applied it to fault diagnosis of the oil-immersed transformer. The results showed that the method was feasible and effective. Yan and Jia [14] proposed a fault recognition algorithm based on optimized multidomain feature SVM. The feature vectors of fault samples were extracted from the time domain, frequency domain, and time-frequency domain. And Laplace fractional algorithm was introduced to filter fault features. Zhong et al. [15] established a diagnosis model based on convolutional neural network transmission learning and SVM and verified the effectiveness of the model through an example. For the accuracy of transformer fault diagnosis, Huang et al. [16] proposed a diagnosis method based on an improved gray wolf algorithm and SVM. The differential evolution mechanism was introduced into the gray wolf optimization algorithm to improve its performance, and then the SVM optimized by the improved gray wolf algorithm was used for fault diagnosis of the transformer.

Equipment fault diagnosis under the condition of incomplete data also has certain research and development. Zhang and Dong [17] proposed an online nonimputation reasoning method based on mixed Gaussian output for fault detection and identification and proved that the method can accurately identify the fault. Mao et al. [18] studied the bearing fault diagnosis with unbalanced data and constructed an online fault prediction method based on an extreme learning machine. The simulation experiment showed that the method can obtain high fault diagnosis accuracy. Liu et al. [19] proposed a Bayesian network parameter learning method based on BPNN and maximum likelihood estimation to solve the problem of solar-assisted heat pump fault diagnosis under the condition of lack of small sample data and lack of expert knowledge. BP neural network was used to predict and fill in the missing sample data, and the effectiveness of the method was verified by simulation. Chen et al. [20] constructed a fault diagnosis model of missing data based on transfer learning for the fault diagnosis problem with too small complete sample size, an appropriate migration learning mechanism was established to improve the accuracy of fault diagnosis, and the effectiveness of this method was verified by data. Zhao et al. [21] constructed a rolling bearing fault diagnosis model based on normalized CNN under unbalanced data and eliminated the difference of feature distribution by batch normalization. The experimental results showed that the model has a good diagnosis effect and robustness for rolling bearing fault diagnosis under unbalanced data. Qian and Li [22] established a kind of unbalance robust network for bearing fault diagnosis, which was used to solve the class imbalance problem in the feature extraction stage and classification stage, and the method was verified by simulation analysis. Zhang et al. [23] proposed to use the deep learning method to solve the problem of fault diagnosis when the data was unbalanced and established a deep generated countermeasure network to generate false samples to balance the sample data. Simulation experiments showed that the proposed method has a better effect on fault diagnosis under unbalanced data.

Collecting sample data in the field of fault diagnosis, a large number of fault sample data cannot be obtained because equipment may not operate normally due to the existence of faults. Presently, most of the research on equipment fault diagnosis is based on complete data set, the research on equipment fault diagnosis under incomplete data is less, and there are some problems such as complex

diagnosis process, long diagnosis time, and unsatisfactory accuracy.

Small sample data missing can not only increase the difficulty of data analysis but also greatly affect the accuracy of the equipment failure diagnosis. For most of equipment failure diagnosis under data missing, it needs a large number of failure sample data to obtain more accurate diagnosis results. Actually, due to equipment aging or human error, a large number of sample data cannot be collected, and there is sample data missing. Thus, the objective of this paper is to propose a novel small sample data missing filling method based on GA-SVR to improve the equipment failure diagnosis effect.

For equipment fault diagnosis, ANN needs a large number of failure samples to train the neural network, but it is difficult to obtain enough failure data in practical application. Additionally, the neural network has the disadvantages of slow convergence, overfitting, and ease to fall into the local optimal value. These will have an adverse impact on the diagnostic accuracy of equipment. Actually, equipment may not operate normally due to failure. And it is unable to obtain a large number of failure sample data. SVR needs less training samples and has high model accuracy. Thus, it is suitable for equipment fault diagnosis in the case of small samples. The advantages of GA lie in its fast optimization speed, good effect, and strong global search ability, and it is not easy to fall into the local optimal solution. Thus, it is used to optimize the key parameters of SVR. In this paper, first, the GA-SVR model is trained by using other data of the variable to which the missing data belongs, and the single-variable prediction method can be obtained. The correlation analysis is used to reconstruct the training set, and the GA-SVR is trained by using the data of the variables related to the missing data to obtain the multivariate prediction method. Then, the dynamic weight is presented to combine the single-variable prediction method with the multiple-variable prediction method based on certain principles, and the missing data are filled with the combined prediction methods. The filled data are used as input of GA-SVM to diagnose equipment failure. Finally, a case study is given to verify the applicability and effectiveness of the proposed method.

This paper aims to develop a new method for equipment health diagnosis. The paper is organized as follows. In section 2, the basic theories of SVR and GA are introduced. Section 3 develops a novel GA-SVR. In Section 4, a case study for equipment health diagnosis with small sample data missing is analyzed and discussed. Finally, conclusions are drawn in Section 5.

2. Theoretical Background

2.1. Support Vector Regression. For the support vector regression (SVR), it is to use the given sample data to fit a continuous function which can reflect the relationship between input and output. In the case that the sample is linear and inseparable, SVR uses a nonlinear transformation to map the data set to a high-dimensional space and carries out regression fitting in this space to establish the continuous function with the minimum loss function.

The key parameters of SVR include insensitive loss function ε, radial basis function parameter σ, and penalty factor C. ε represents the insensitive region width and plays a decisive role in the number of support vectors and the generalization ability of the model. σ determines the complexity of sample mapping space. The larger σ means that it is difficult to obtain high regression accuracy. The smaller σ means that the regression accuracy is high and the generalization ability is poor. C represents the penalty degree for samples with an error greater than ε. The larger C indicates that the penalty for samples is large. Although the training accuracy can be improved, the generalization ability of the model is poor. The smaller C shows that the penalty for samples is very small, and it will cause a large training error. These three key parameters determine the performance of SVR; thus, it is necessary to optimize these parameters to improve the prediction effect of SVR.

2.2. Genetic Algorithm. Genetic algorithm (GA) is a kind of heuristic optimization technology. GA searches from the initial population generated randomly, and the individuals in the population evolve through selection, crossover, and mutation based on the fitness function until the iteration termination condition is met, and the optimal solution is output.

The advantages of GA include fast optimization speed and strong global searchability, and it is not easy to fall into the local optimal solution. It is widely used in various optimization problems such as parameter optimization and path optimization.

The basic procedure of GA is as follows:

Step 1. The chromosome needs to be coded to determine the initial population

Step 2. The fitness function is described to evaluate the fitness value of individuals

Step 3. The new species group is generated by selection, crossover, and mutation

Step 4. The individuals satisfied the termination iteration condition that can be retained

Step 5. The decoding outputs the global optimal solution

In this paper, for the problem of equipment health diagnosis, SVR is used to predict and fill the missing data. But the values of kernel function parameter σ, penalty factor C, and insensitive loss function ε in SVR are particularly important. Thus, the set of key parameters (C, σ, ε) of SVR can be regarded as a population, and the key parameters of SVR can be optimized by GA to improve the prediction performance of SVR.

3. Equipment Health Diagnosis Based on GA-SVR

3.1. Support Vector Regression Optimized by Genetic Algorithm. SVR is obtained by introducing insensitive loss function into SVM. It is usually used to solve regression

fitting problems and seek a regression function representing the relationship between input and output.

For the given data set $\{x_i, y_i\}$, $i = 1, 2, \ldots, N$, where $x_i \in R^n$ is the input sample, and $y_i \in R$ is the output expected value. Assume that SVR maps samples to a high-dimensional space by nonlinear transformation $\phi(*)$ to establish the regression function, and it is as follows:

$$f(x) = w \cdot \phi(x) + b, \tag{1}$$

where w and b are regression function coefficients. And insensitive loss function ε is introduced and defined as

$$L_\varepsilon(f(x), y) = \begin{cases} |y - f(x)| - \varepsilon, & |y - f(x)| \geq \varepsilon \\ 0, \text{ other} \end{cases} \tag{2}$$

Thus, the objective function can be defined as $\min (1/2)\|w\|^2$, and the constraints are

$$\begin{cases} y_i - w \cdot x_i - b \leq \varepsilon, \\ w \cdot x_i + b - y_i \leq \varepsilon, \end{cases} \quad i = 1, \ldots, N. \tag{3}$$

The relaxation factors ξ_i and ξ_i^* are introduced under the condition of allowing the fitting error; then, the objective function is

$$\min\left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)\right),$$

$$\text{S.T.} \begin{cases} y_i - w \cdot x_i - b \leq \varepsilon + \xi_i, \\ w \cdot x_i + b - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0, \end{cases} \tag{4}$$

where $C > 0$ is the penalty factor, and it is used to control the punishment for errors exceeding ε. By introducing the Lagrange multiplier α_i and α_i^*, then the above problem is transformed into its dual problem.

$$\max\left\{\sum_{i=1}^{N}(\alpha_i^* - \alpha_i)y_i - \sum_{i=1}^{N}(\alpha_i^* + \alpha_i)\varepsilon - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i^* - \alpha_i)\left(\alpha_j^* - \alpha_j\right)K\left(x_i, x_j\right)\right\},$$

$$\text{S.T.} \left\{\sum_{i=1}^{N}\left(\alpha_i^* - \alpha_i\right) = 0, 0 \leq \alpha_i \leq C, 0 \leq \alpha_i \leq C, i = 1, 2, \ldots, N, \right. \tag{5}$$

where $K(x_i, x_j) = \phi(x_i)\phi(x_j)$ is the kernel function. By solving equation (5), the regression fitting function can be obtained as follows:

$$f(x) = \sum_{i=1}^{N}\left(\alpha_i - \alpha_i^*\right)K\left(x_i, x_j\right) + b. \tag{6}$$

For the selection of the SVR kernel function, the RBF kernel function is used in this paper, and its parameter $\sigma > 0$ is the kernel function width factor. It has an important influence on the regression prediction effect of SVR.

The small sample data missing has a great influence on the equipment diagnosis results; thus, this paper uses SVR to execute regression fitting for the missing data. However, the key parameters C, σ, and ε have a great influence on the regression prediction accuracy of SVR. GA is used to optimize C, σ, and ε to improve the prediction performance of SVR for missing data.

The optimization process of C, σ, and ε by GA can be shown in Figure 1, and the specific operation steps are as follows:

Step 1. Parameter initialization: initialize GA parameters and C, σ, and ε; any group (C, σ, ε) represents an individual in GA.

Step 2. Fitness value calculation: in order to evaluate the advantages and disadvantages of GA in selecting SVR parameters, the K-fold cross-validation method is used to take the mean value of K-th root mean square error as the fitness value of an individual, and the calculation of fitness value is as follows:

$$F = \frac{1}{K}\sum_{i=1}^{K}\sqrt{\frac{\sum_{j=1}^{n}(y - \hat{y})^2}{n}}. \tag{7}$$

Step 3. Terminating iteration: if the condition of terminating iteration has not been reached, the selection, crossover, and mutation will be carried out to generate a new group; then, go back to Step 2 to continue iteration.

Step 4. Output optimal values: the optimal values of C, σ, and ε are output after completing iteration and obtain the GA-SVR model.

3.2. Combination Prediction Filling Based on GA-SVR

3.2.1. Single-Variable Prediction Filling Based on GA-SVR. The monitoring data of equipment operation status is mostly time series. It is a series of monitoring values X_t^q obtained by multiple sensors in a time sequence where $t\,(t = 1, 2, \ldots, n)$ represents t-th time point, $q\,(q = 1, 2, \ldots, m)$ denotes the q-th sensor, and X_t^q means the monitoring data value corresponding to the q-th sensor at the t-th time point.

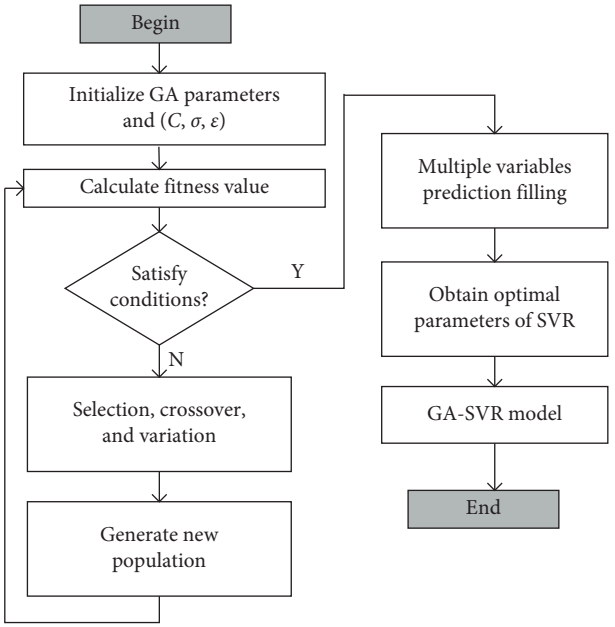


Figure 1: The flow chart of SVR parameters optimized by GA.

Using GA-SVR to predict the single variable of missing data is to train GA-SVR by using other data of variables with missing data as input to predict the value of missing data.

First, let the length of the missing data segment be l, and determine the variable q of the missing data. The n-l-1 data values in the q-th variable dimension are selected as the input of GA-SVR, and the remaining data value is used as the output to train GA-SVR. Then, the trained GA-SVR model is used to predict missing data, and the single-variable prediction results can be obtained.

3.2.2. Multiple-Variable Prediction Filling Based on GA-SVR. This paper uses GA-SVR to predict the missing data. The data related to the variable dimension containing missing data is used as input to train the GA-SVR model and predict the value of missing data.

First, the correlation analysis is used to find the other variables related to the variable q to form the training set $X_1, \cdots, X_i, \cdots, X_k$. X_t represents the monitoring value at t-th time point. The correlation coefficient R is used to evaluate the correlation among the variables. If the correlation coefficient $R \geq 0.8$, it indicates that the two variables are strongly correlated. The correlation coefficient R is calculated as follows:

$$R = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2 \sum_{i=1}^{n} (y_i - \overline{y})^2}}. \tag{8}$$

The monitoring data from 1-st to k-th time point can be used to execute correlation analysis. And the GA-SVR is trained with the monitoring data values at remaining n-k time points as the input and the data values at a time point where the missing data belongs to as the output. Then, the trained GA-SVR model is used to predict the missing data and obtain the multivariable prediction results.

3.2.3. Dynamic Weight Combination Prediction Filling Based on GA-SVR. In order to improve the accuracy of missing data prediction, reduce the deviation between the predicted value and the actual value, and improve the effectiveness of equipment fault diagnosis, a dynamic weight combination prediction method based on GA-SVR is established to fill the missing data. GA-SVR is used to make a single-variable prediction and multiple-variable predictions, respectively, and then the dynamic weight combination of single-variable prediction and multiple-variable prediction results is obtained. The combined prediction results are used to fill in the missing data to obtain complete data set.

Root mean square error (RMSE) can describe the deviation between the predicted value and the actual value. Thus, RMSE is used to evaluate the quality of the prediction results. The smaller RMSE represents the better prediction effect of missing data. The root mean square error is expressed as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^{n} (y_i - \widehat{y}_i)^2}{n}}, \tag{9}$$

where y_i is the actual value, $\widehat{y}_i$ is the predicted value, and n is the prediction times.

The weight value of single-variable prediction results and multiple-variable prediction results in combination forecasting depends on their root mean square error difference. The root mean square error is smaller, and the weight is greater. Based on equation (10), the prediction result of missing data can be obtained and it is followed as equation (11).

$$\begin{cases} w_1 = \dfrac{k}{R_1}, \\ w_2 = \dfrac{k}{R_2}, \\ w_1 + w_2 = 1, \\ y_i^* = w_1 \widehat{y}_{1i} + w_2 \widehat{y}_{2i}, \end{cases} \tag{10}$$

$$y_i^* = \frac{R_2}{R_1 + R_2} \widehat{y}_{1i} + \frac{R_1}{R_1 + R_2} \widehat{y}_{2i}, \tag{11}$$

where $\widehat{y}_{1i}$ and $\widehat{y}_{2i}$ denote single-variable prediction results and multiple-variable prediction results, respectively. R_1 and R_2 are the RMSE values corresponding to single-variable prediction and multiple-variable prediction, respectively. y_i^* is the final missing data filling values.

The chart of combination prediction based on GA-SVR can be seen in Figure 2.

3.3. Equipment Failure Diagnosis Procedure. For the problem of equipment fault diagnosis under the condition of small sample data missing, GA-SVR is used to fill the missing data, and the complete data after filling is used as the input of SVM to realize the fault diagnosis of equipment. The

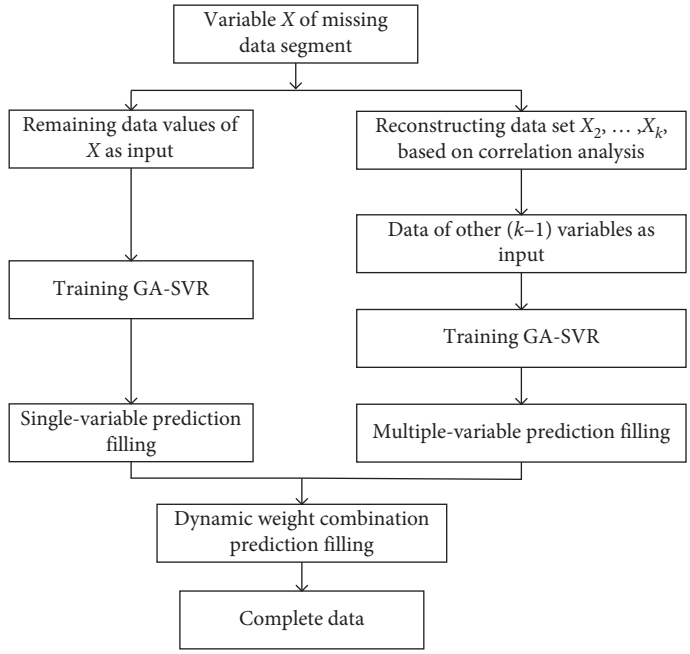


Figure 2: The flow chart of combination prediction based on GA-SVR.

fault diagnosis flow chart based on SVR under the condition of small sample data missing can be shown in Figure 3, and the specific fault diagnosis scheme can be shown as follows:

Step 1. The other data of the variable to which the missing data belongs is used to train the GA-SVR model to obtain the single-variable prediction filling result that can be obtained.

Step 2. Find out the variables related to the variables of missing data by correlation analysis, and the data of these variables can be used to train the GA-SVR model. The multiple-variable prediction filling results can be obtained.

Step 3. Based on equation (11), the single-variable prediction results and the multivariate prediction results are combined to obtain the combined prediction results, and the missing data are filled to obtain the complete data.

Step 4. The complete data is divided into training sample data set and test sample data set, and SVM is trained and tested, respectively, to obtain the fault diagnosis results of equipment.

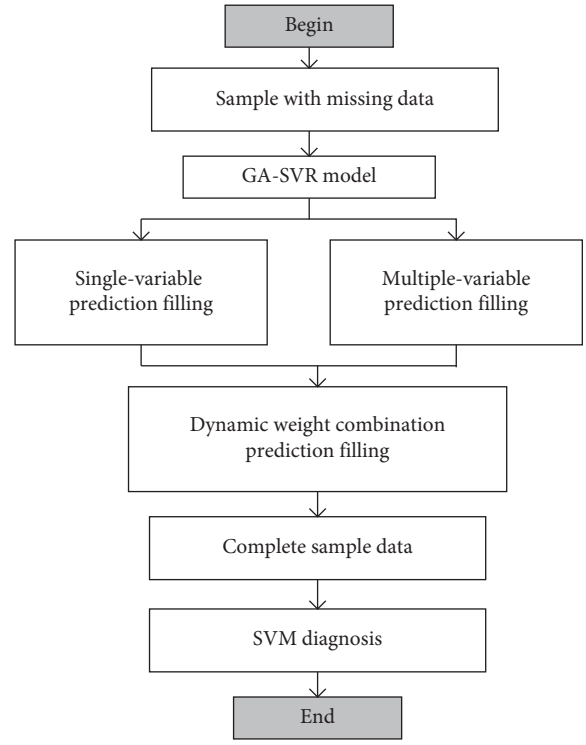


Figure 3: Equipment fault diagnosis scheme based on GA-SVR with small sample data missing.

4. Case Study

4.1. Experimental Setup and Data Acquisition. To validate the proposed methods, a real-world case is studied. In this case study, the long-term wear test experiments were conducted at a research laboratory facility. In the test experiments, three pumps (A, B, and C) were worn by running them using oil containing dust. Each pump experienced four states: Baseline state, Degradation state, Degradation state, and Failure state. The degradation stages in this hydraulic pump wear test case study correspond to different stages of flow loss in the pumps. As the flow rate of a pump clearly indicates the pump's health state, the degradation stages corresponding to different degrees of flow loss in a pump were defined as the health states of the pump in the test [24, 25].

The vibration signals were collected from pump accelerometers that were positioned parallel to the axis of the

swash plate swivel axis and data was continuously sampled. Figure 4 shows the schematic diagram of the experimental setup. The pump used for testing in the experiments was a Back Hoe Loader: a 74 cc/rev variable displacement pump. The data was collected at a sample rate of 60 kHz with antialiasing filters from accelerometers designed to have a usable range of 10 kHz. In many cases, the most distinguished information is hidden in the frequency content of signals. So, the time-frequency representation of signals is needed. In this case study, the signals were processed using a wavelet packet with Daubechies wavelet 10 (db10), and five decomposition levels as the db10 wavelet provide the most effective way to capture the fault information in the pump vibration data. The coefficients obtained by the wavelet packet decomposition were used as the inputs.

There are 80 groups of experimental data for Pumps A, B, and C, respectively. Each group of data contains 32 variables (32 sensors). In this paper, the monitoring data of the 3-th sensor is taken as the experimental object, and the monitoring data from the 75-th to 80-th time point is deleted to simulate the missing situation of small sample data. The single-variable prediction, multiple-variable prediction, and dynamic weight combination prediction based on GA-SVR are used to fill the missing data, and the filling effect and the diagnosis effect after filling are compared.

4.2. Reconstruction Training Set. The multiple-variable prediction model selects monitoring data from sensors having a strong correlation with Sensor 3 as the training set to predict the missing data value. Based on equation (8), the correlation coefficients between Sensor 3 and other sensors are calculated in Pumps A, B, and C, respectively. If the correlation coefficient $R \geq 0.8$, then the sensor and Sensor 3 have a strong correlation; thus, the training set can be reconstructed, as shown in Tables 1–3. The reconstructed training sample is only 6-dimensional. It can reflect the characteristics of the original data, reduce the amount of calculation, and shorten the prediction time.

4.3. Result Analysis of Missing Data Filling. In order to evaluate the filling effect of the proposed dynamic weight combination prediction method based on GA-SVR, the missing values in Pumps A, B, and C are predicted by single-variable prediction, multiple-variable prediction, and dynamic weight combination prediction by using GA-SVR, respectively. And the filling effects are compared.

The parameters of GA are set as follows: the population size is 20, and the maximum iteration number is 100. The key parameters of SVR are $0.1 \leq C \leq 1000$, $0.01 \leq \sigma \leq 100$, and $0.01 \leq \varepsilon \leq 1$. The root mean square error (RMSE) and mean absolute percentage error (MAPE) are used as the evaluation indexes for the filling effect of missing data. MAPE is as follows:

$$\text{MAPE} = \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right| \times \frac{100\%}{n}, \tag{12}$$

where y_i is the actual value and $\hat{y}_i$ is the predicted value.

Tables 4–6 show the predicted filling values of missing data of Pumps A, B, and C based on GA-SVR, respectively. Figures 5–7 show the missing data fitting curves of three prediction methods based on GA-SVR for Pumps A, B, and C, respectively.

From Figures 5–7, it can be intuitively seen that the simulation results of the three data sets are basically consistent. The fitting curve of dynamic weight combination prediction is more consistent with the actual value curve than that of single-variable prediction and multiple-variable prediction. It indicates that the effect of the dynamic weight combination prediction method is better than that of single-variable prediction and multiple-variable prediction.

In order to evaluate the effect of equipment fault diagnosis under the small sample data missing based on the proposed GA-SVR, the proposed GA-SVR prediction model is compared with the standard SVR prediction model and BP neural network prediction model (BPNN). The key parameters of SVR are selected by grid search cross-validation method, $0.1 \leq C \leq 1000$, $0.01 \leq \sigma \leq 100$, and $0.01 \leq \varepsilon \leq 0\,1$. For the single-variable prediction of missing data, the input layer of BPNN is 1, the output layer is 1, and the number of hidden layers is 3. For the multiple-variable prediction of missing data, the input layer of BPNN is 6, the output layer is 1, and the number of hidden layers is 5. The maximum iteration times are set to 100, the error accuracy is 0.002, the learning rate is 0.1, and the activation function is a sigmoid type function.

Tables 7–9 show the filling effect of missing data of Pumps A, B, and C for three different prediction models, respectively. It can be seen from Tables 7–9 that the RMSE and MAPE values of dynamic weight combination prediction are the smallest compared with single-variable prediction and multiple-variable prediction for different prediction modes of the same prediction model. For the same prediction mode of different prediction models, the RMSE and MAPE values of the proposed GA-SVR model are the minimum. Thus, the proposed dynamic weight combination prediction of missing data based on GA-SVR has the best filling effect on missing data.

4.4. Result Analysis of Equipment Failure Diagnosis. In order to compare the effects of different missing data prediction models and prediction modes on equipment fault diagnosis, the complete data filled with missing data is used for equipment fault diagnosis. 50 groups of Pumps A, B, and C data sets are randomly selected as training samples, and the remaining 30 groups are used as test samples.

Tables 10–12 show the influence of three different missing data filling models of GA-SVR, SVR, and BPNN and three prediction filling modes on the fault diagnosis effect of Pumps A, B, and C, respectively. It can be seen from Tables 10–12 that the dynamic weight combination prediction filling mode has the highest diagnosis accuracy rate and shorter time compared with single-variable prediction filling mode and multiple-variable prediction filling mode under the same prediction model. For the same prediction mode, the fault diagnosis rate based on GA-SVR is the

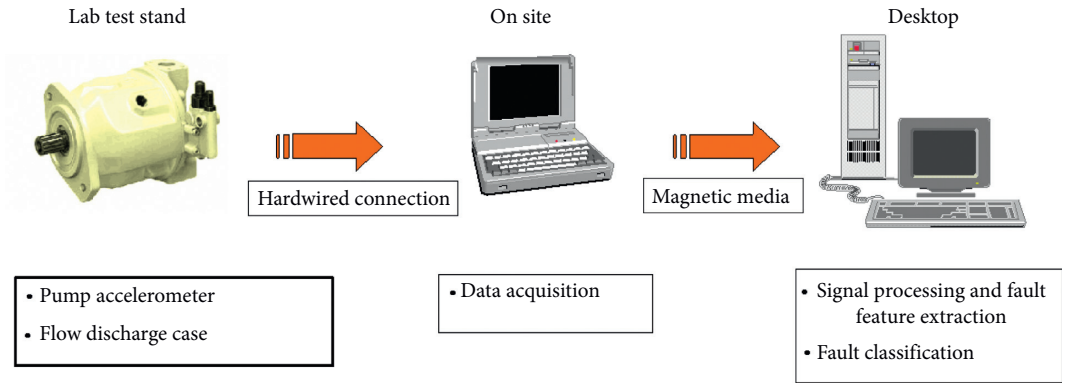


Figure 4: Schematic diagram of the experimental setup.

Table 1: Sensors having a strong correlation with Sensor 3 in hydraulic Pump A.

	CH2	CH5	CH7	CH13	CH16	CH32
R	0.826	0.872	0.908	0.911	0.956	0.858

Table 2: Sensors having a strong correlation with Sensor 3 in hydraulic Pump B.

	CH2	CH5	CH7	CH13	CH16	CH32
R	0.819	0.863	0.895	0.902	0.938	0.838

Table 3: Sensors having a strong correlation with Sensor 3 in hydraulic Pump C.

	CH2	CH5	CH7	CH13	CH16	CH32
R	0.821	0.867	0.901	0.907	0.944	0.843

Table 4: Prediction results of missing data based on GA-SVR for Pump A.

Actual value	Single-variable predicted value	Multiple-variable predicted value	Dynamic weight combination predicted value
16.9640	16.9023	17.0052	16.9502
16.8942	16.8425	16.9732	16.9033
16.7349	16.7745	16.6997	16.7397
16.6608	16.7177	16.6369	16.6801
16.6291	16.6791	16.6002	16.6424
16.7138	16.7330	16.7265	16.7300

Table 5: Prediction results of missing data based on GA-SVR for Pump B.

Actual value	Single-variable predicted value	Multiple-variable predicted value	Dynamic weight combination predicted value
15.1519	15.3855	14.9987	15.2222
14.2496	13.7533	14.4974	14.0675
12.8249	12.5942	13.0492	12.7863
12.9940	12.5854	13.1238	12.8128
12.3819	12.5935	11.9923	12.3396
12.4991	12.8678	12.2324	12.5995

Table 6: Prediction results of missing data based on GA-SVR for Pump C.

Actual value	Single-variable predicted value	Multiple-variable predicted value	Dynamic weight combination predicted value
9.4516	9.3048	9.5537	9.4079
9.3964	9.3058	9.4623	9.3706
9.7349	9.6812	9.7615	9.7145
9.1048	9.1879	9.0531	9.1321
9.5237	9.5981	9.4552	9.5389
9.6634	9.6289	9.6801	9.6501

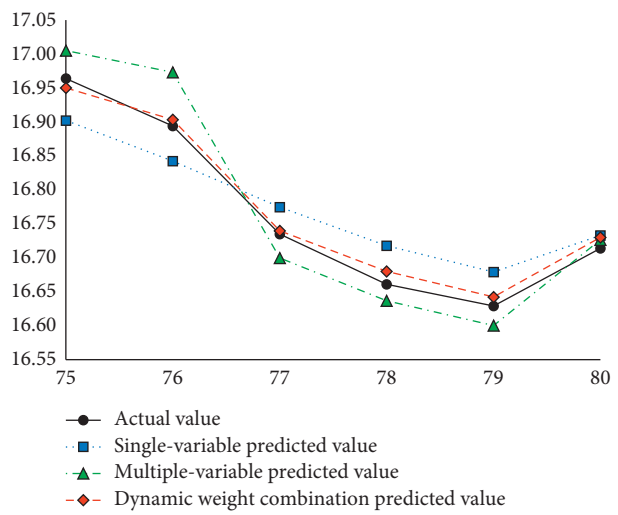


Figure 5: The fitting curve of missing data based on GA-SVR for Pump A.

highest compared with SVR and BPNN, and the diagnosis time is shorter than that of BPNN. And the diagnosis time is longer than SVR, but the difference is not significant.

Generally, the missing data filling method of dynamic weight combination prediction based on GA-SVR can obtain the best fault diagnosis effect. It can be concluded that the proposed failure diagnosis method based on GA-SVR under the condition of small sample missing data is effective for Pumps A, B, and C and has certain universality.

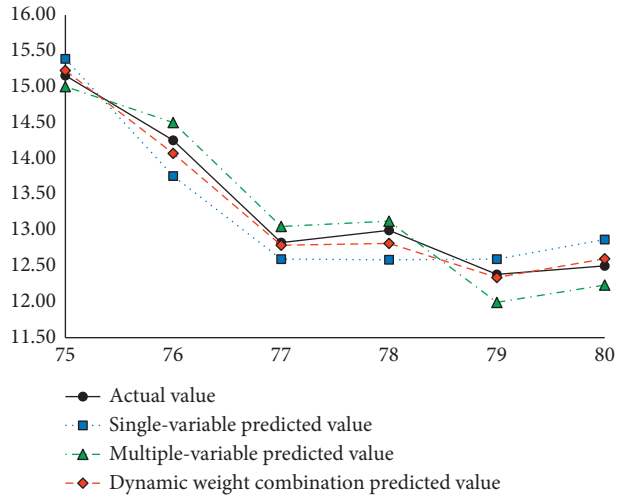


Figure 6: The fitting curve of missing data based on GA-SVR for Pump B.

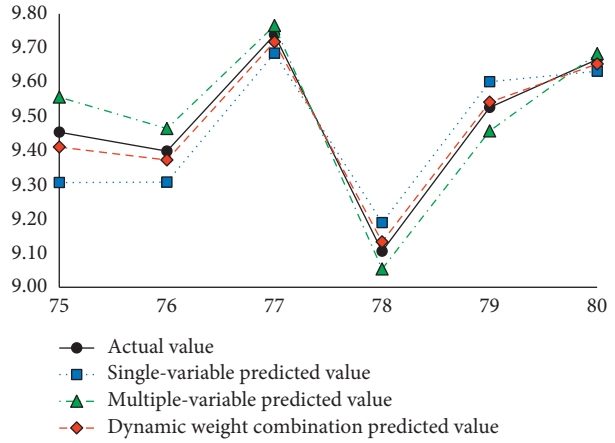


Figure 7: The fitting curve of missing data based on GA-SVR for Pump C.

Table 7: Prediction effect of missing data of Pump A for three different prediction models.

	Single-variable prediction		Multiple-variable prediction		Dynamic weight combination prediction	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
GA-SVR	0.0486	0.28	0.0423	0.22	0.0138	0.08
SVR	0.0737	0.40	0.0500	0.25	0.0303	0.16
BPNN	0.0920	0.52	0.0644	0.36	0.0547	0.28

Table 8: Prediction effect of missing data of Pump B for three different prediction models.

	Single-variable prediction		Multiple-variable prediction		Dynamic weight combination prediction	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
GA-SVR	0.3420	2.44	0.2500	1.80	0.1185	0.76
SVR	0.6547	2.90	0.3989	2.13	0.2158	1.12
BPNN	0.8832	3.28	0.5150	2.82	0.2990	1.98

Table 9: Prediction effect of missing data of Pump C for three different prediction models.

	Single-variable prediction		Multiple-variable prediction		Dynamic weight combination prediction	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE (%)
GA-SVR	0.0878	0.85	0.0621	0.59	0.0263	0.26
SVR	0.1439	1.69	0.1219	1.33	0.0498	0.73
BPNN	0.2293	2.12	0.1580	1.83	0.0724	1.20

Table 10: Failure diagnosis effect of different missing data filling models for Pump A.

	Single-variable prediction filling mode		Multiple-variable prediction filling mode		Dynamic weight combination prediction filling mode	
	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)
GA-SVR	83.33	20.84	90.00	41.20	96.67	41.64
SVR	80.00	21.33	83.33	39.86	90.00	41.50
BPNN	76.67	50.8	90.00	87.23	93.33	88.92

Table 11: Failure diagnosis effect of different missing data filling models for Pump B.

	Single-variable prediction filling mode		Multiple-variable prediction filling mode		Dynamic weight combination prediction filling mode	
	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)
GA-SVR	86.67	16.43	93.33	23.45	100.00	24.29
SVR	83.33	14.50	90.00	22.76	96.67	23.40
BPNN	76.67	31.80	86.67	59.80	93.33	61.02

Table 12: Failure diagnosis effect of different missing data filling models for Pump C.

	Single-variable prediction filling mode		Multiple-variable prediction filling mode		Dynamic weight combination prediction filling mode	
	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)	Accuracy (%)	Time (s)
GA-SVR	86.67	9.43	93.33	13.45	96.67	14.23
SVR	83.33	8.55	86.67	10.98	93.33	11.45
BPNN	80.00	15.78	83.33	21.50	90.00	23.27

5. Conclusion

In this paper, for the problem that small sample data missing will affect the effect of equipment failure diagnosis, a novel missing data filling method based on GA-SVR is proposed to improve the effect of the equipment failure diagnosis. First, the single-variable prediction is carried out for the missing data. And the training set is reconstructed by correlation analysis. Meanwhile, the multiple-variable prediction is carried out based on GA-SVR. Then, the dynamic weight is presented to combine the single-variable prediction results and the multiple-variable prediction results to fill in the missing data. Finally, the complete data obtained by filling missing data is used as input, and GA-SVM is used to diagnose the equipment failure.

By the case study, the proposed GA-SVR model is compared with SVR and BPNN to predict the filling effect of missing data of Pumps A, B, and C, respectively. And the failure diagnosis effect based on the complete data after the filling is compared. It can be shown that the proposed dynamic weight combination prediction method based on GA-SVR has the best missing data filling effect and failure diagnosis effect. And the effectiveness and universality of this proposed method under the condition of small sample data missing can be verified.

Data Availability

The underlying data supporting the results of our study can be obtained, including, where applicable, hyperlinks to publicly archived datasets analyzed or generated during the study, upon request to the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by grants from the National Natural Science Foundation of China (Nos. 71632008, 71840003, and 51875359), Natural Science Foundation of Shanghai (Nos. 19ZR1435600 and 20ZR1428600), and

Humanity and Social Science Planning Foundation of the Ministry of Education of China (No. 20YJAZH068).

References

[1] Z. Husain, "Fuzzy logic expert system for incipient fault diagnosis of power transformers," *International Journal on Electrical Engineering and Informatics*, vol. 10, no. 2, pp. 300–317, 2018.

[2] T. Berredjem and M. Benidir, "Bearing faults diagnosis using fuzzy expert system relying on an Improved Range Overlaps and Similarity method," *Expert Systems with Applications*, vol. 108, pp. 134–142, 2018.

[3] A. Cheriet, A. Bekri, A. Hazzab, and H. Gouabi, "Expert system based on fuzzy logic: application on faults detection and diagnosis of DFIG," *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 9, no. 3, pp. 1081–1089, 2018.

[4] X. Xu, X. Yan, C. Sheng et al., "A belief rule based expert system for fault diagnosis of marine diesel engines," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 2, pp. 656–672, 2020.

[5] C. H. Xing, F. T. Xu, Z. Y. Yao et al., "A fault diagnosis method of reciprocating compressor based on sensitive feature evaluation and artificial neural network," *High Technology Letters*, vol. 21, no. 4, pp. 422–428, 2015.

[6] R. Yang, M. Huang, Q. Lu, and M. Zhong, "Rotating machinery fault diagnosis using long-short-term memory recurrent neural network," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 228–232, 2018.

[7] R. S. Gunerkar, A. K. Jalan, and S. U. Belgamwar, "Fault diagnosis of rolling element bearing based on artificial neural network," *Journal of Mechanical Science and Technology*, vol. 33, no. 2, pp. 505–511, 2019.

[8] C. Wu, P. Jiang, C. Ding, F. Feng, and T. Chen, "Intelligent fault diagnosis of rotating machinery based on one-dimensional convolutional neural network," *Computers in Industry*, vol. 108, pp. 53–61, 2019.

[9] Y. Han, B. Tang, and L. Deng, "An enhanced convolutional neural network with enlarged receptive fields for fault diagnosis of planetary gearboxes," *Computers in Industry*, vol. 107, pp. 50–58, 2019.

[10] J. Huang and X. G. Hu, "Support vector machine with genetic algorithm for machinery fault diagnosis of high voltage circuit breaker," *Measurement*, vol. 44, no. 6, pp. 1018–1027, 2011.

[11] S. W. Fei and X. B. Zhang, "Fault diagnosis of power transformer based on support vector machine with genetic algorithm," *Expert Systems with Applications*, vol. 36, no. 8, pp. 11352–11357, 2009.

[12] D. Yang, Y. Liu, S. Li, X. Li, and L. Ma, "Gear fault diagnosis based on support vector machine optimized by artificial bee colony algorithm," *Mechanism and Machine Theory*, vol. 90, no. 90, pp. 219–229, 2015.

[13] Y. Zhang, H. Wei, R. Liao, Y. Wang, L. Yang, and C. Yan, "A new support vector machine model based on improved imperialist competitive algorithm for fault diagnosis of oil-immersed transformers," *Journal of Electrical Engineering and Technology*, vol. 12, no. 2, pp. 830–839, 2017.

[14] X. Yan and M. Jia, "A novel optimized SVM classification algorithm with multi-domain feature and its application to fault diagnosis of rolling bearing," *Neurocomputing*, vol. 313, pp. 47–64, 2018.

[15] S.-s. Zhong, S. Fu, and L. Lin, "A novel gas turbine fault diagnosis method based on transfer learning with CNN," *Measurement*, vol. 137, pp. 435–453, 2019.

[16] X. Huang, X. Huang, B. Wang, and Z. Xie, "Fault diagnosis of transformer based on modified grey wolf optimization algorithm and support vector machine," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 15, no. 3, pp. 409–417, 2020.

[17] Z. Zhang and F. Dong, "Fault detection and diagnosis for missing data systems with a three time-slice dynamic Bayesian network approach," *Chemometrics and Intelligent Laboratory Systems*, vol. 138, pp. 30–40, 2014.

[18] W. Mao, L. He, Y. Yan, and J. Wang, "Online sequential prediction of bearings imbalanced fault diagnosis by extreme learning machine," *Mechanical Systems and Signal Processing*, vol. 83, pp. 450–473, 2017.

[19] Z. Liu, Y. Liu, D. Zhang, B. Cai, and C. Zheng, "Fault diagnosis for a solar assisted heat pump system under incomplete data and expert knowledge," *Energy*, vol. 87, pp. 41–48, 2015.

[20] D. Chen, S. Yang, and F. Zhou, "Transfer learning based fault diagnosis with missing data due to multi-rate sampling," *Sensors*, vol. 19, no. 8, p. 1826, 2019.

[21] B. Zhao, X. Zhang, H. Li, and Z. Yang, "Intelligent fault diagnosis of rolling bearings based on normalized CNN considering data imbalance and variable working conditions," *Knowledge-Based Systems*, vol. 199, Article ID 105971, 2020.

[22] W. Qian and S. Li, "A novel class imbalance-robust network for bearing fault diagnosis utilizing raw vibration signals," *Measurement*, vol. 156, Article ID 107567, 2020.

[23] W. Zhang, X. Li, X.-D. Jia, H. Ma, Z. Luo, and X. Li, "Machinery fault diagnosis with imbalanced data using deep generative adversarial networks," *Measurement*, vol. 152, Article ID 107377, 2020.

[24] M. Dong and D. He, "A segmental hidden semi-Markov model (HSMM)-based diagnostics and prognostics framework and methodology," *Mechanical Systems and Signal Processing*, vol. 21, no. 5, pp. 2248–2266, 2007.

[25] K. M. Hancock and Q. Zhang, "A hybrid approach to hydraulic vane pump condition monitoring and fault detection," *Transactions of the ASABE*, vol. 49, no. 4, pp. 1203–1211, 2006.

Hindawi
Shock and Vibration
Volume 2021, Article ID 6699611, 10 pages
https://doi.org/10.1155/2021/6699611

Research Article

Health State Prediction and Performance Evaluation of Belt Conveyor Based on Dynamic Bayesian Network in Underground Mining

Xiangong Li,[1] **Yuzhi Zhang**,[1] **Yu Li**,[1] **Yujie Zhan**,[2] **and Lin Yang**[3]

[1]*School of Mines, China University of Mining and Technology, Xuzhou 221116, China*
[2]*School of Industrial and Business Management, Xuzhou College of Industrial Technology, Xuzhou 221116, China*
[3]*School of Business Administration, Nanjing University of Finance and Economics, Nanjing 210023, China*

Correspondence should be addressed to Xiangong Li; cumtlxg@163.com and Lin Yang; ylnju@126.com

Received 17 December 2020; Accepted 2 January 2021; Published 20 February 2021

Academic Editor: Tangbin Xia

To deal with the problem of weak prediction and performance evaluation capabilities of traditional prediction and evaluation methods, a method of health state prediction and performance evaluation of belt conveyor based on Dynamic Bayesian Network (DBN) is proposed. First, the belt conveyor sensor monitoring data are preprocessed to obtain the feature data set with labels. At the same time, qualitative and quantitative analyses and interval discretization are carried out from belt conveyor fault-causing elements to construct the DBN network. Then, the sample data are applied to the DBN network for training, and the DBN-based prediction and performance evaluation model is established. Finally, taking the real-time monitoring data of a belt conveyor in an underground mine as an example, a DBN-based belt conveyor health prediction and evaluation model is constructed to evaluate and predict the health of the equipment. The results show that the model could identify different operating conditions and failure modes and further improves the prediction accuracy.

1. Introduction

As a key transport equipment in coal production, the belt conveyor is widely used in underground coal mine and main transport roadway on the ground [1, 2]. The working environment of belt conveyor in underground mines is often complex and hazardous, which cause belt faults such as conveyor deviation, sliding, broken belt, spreader, and reducer shaft. The health status condition of the belt conveyor directly affects the workload of the working face and the life safety of the coal mine site operators [3-5]. Therefore, accurate prediction and performance evaluation of belt conveyor health status is the necessary and prerequisite for health management of critical equipment in coal mines [6-8].

Most techniques used for equipment health state prediction and performance evaluation fall into two main categories [9, 10], which are model-based approaches [11, 12] and data-driven approaches [13, 14]. Most model-based approaches require the identification of accurate physical or mathematical models to describe the process of device health state changes. Data-driven model prediction and performance evaluation methods have become an important approach of prediction and performance assessment for complex equipment for its difficulty to determine a specific health state [15–17]. The data-based prediction approaches are mainly based on data fusion and feature extraction of sensor history data of the system or component to obtain a mapping relationship between data and health states [18, 19]. The method is not combined with the a priori knowledge of the device itself and is a more practical method for prediction and evaluation operations based on the existing collected data and mining the implicit correlation information in the data through various analytical processing methods [20-22]. In general, most of the existing studies focus on prediction and assessment under a single health state and failure mode, ignoring a certain extent of influence of environmental and operational conditions [23–25]. In practical engineering applications, traditional prediction and evaluation methods fall short in massive monitoring data, which in turn affects the effectiveness of prediction [26–28].

Based on this, a method was proposed for health state prediction and performance evaluation of belt conveyor based on DBN, carries out quantitative evaluation on the health state of belt conveyor, and predicts the change in the health state of the belt conveyor in the future time slice. First, the historical monitoring data and patrol statistical data on the belt conveyor sensor of the coal mine monitoring system are collected, and the health state prediction and performance evaluation indexes of the belt conveyor are mined. The indexes are discretized to reduce the influence of time parameters on the training results. Then, it analyses the process of DBN network learning and reasoning. Finally, the preprocessed data are input to DBN network for training, and a DBN model for health state prediction and performance evaluation of belt conveyor is established. Experiments show that the proposed approach could effectively solve the shortcoming of traditional methods in data processing, and it has strong feasibility and practicability which could improve the equipment performance in engineering application.

2. Data-Driven DBN Prediction and Performance Evaluation Method

In this section, the basic principles of DBNs which include basic assumptions, structure learning, parameter learning, inference, and prediction are introduced. Based on this, a DBN prediction and performance evaluation method is proposed.

2.1. Basic Assumptions of DBN. Bayesian network (BN) is a system model at a given time, which is used to model a system in some states of equilibrium [28]. A Bayesian network can be defined as

$$\mathrm{BN} = (G, \theta), \tag{1}$$

where G is the directed acyclic graph of the joint probability distribution over node Z and θ is a parameter in the network, and the joint probability distribution of Z is as follows:

$$P(Z_1, Z_2, \ldots, Z_n) = \prod_{i=1}^{n} P(Z_i | Pa(Z_i)). \tag{2}$$

DBN is a dynamic Bayesian network that simulate the effects of changes in the network over time, reflecting changes in the health of the system at different times. To describe this specific process, some assumptions need to be made as follows:

(1) Steady-state assumption: the conditional probability of a network node is the same for all time slices t, and the transfer probability of any two neighbouring time slices is the same.

(2) The first-order Markovian hypothesis: the current state depends only on the state of the previous moment, independent of the state of the previous moment, i.e.,

$$P\left(Z^t | Z^0, Z^1, \ldots, Z^{t-1}\right) = P\left(Z^t | Z^{t-1}\right). \tag{3}$$

2.2. Structural Learning of DBN. Based on the above assumptions, a DBN can be defined as $(B_0, B \longrightarrow)$, where B_0 is the joint probability distribution specifying the initial state $Z[0]$ of the variable and $B \longrightarrow$ refers to the transfer probability $P(Z^t | Z^{t-1})$ (which holds for all t) on variables $Z[0]$and $Z[1]$.The Bayesian network formula for two neighbouring time slices is shown in the following equation:

$$P(Z_t | Z_{t-1}) = \prod_{i=1}^{N} P\left(Z_t^i | Pa\left(Z_t^i\right)\right), \tag{4}$$

where Z_t^i is the value of the i-th variable, moment t, and $Pa(Z_t^i)$ is the parent of Z_t^i.

The process in the DBN is fixed and the structure is repeated after the second time slice, and the variable $t = 2, 3, \ldots, T$in the slice DBN is kept constant, so that the system can be expressed by only two adjacent slices (i.e., the first and second time slice) and a finite number of parameters can be used to simulate the unbounded sequence length. The probability distribution of the time slice sequences obtained by expanding the 2TBN network is shown as follows:

$$P(Z_{1:T}) = \prod_{t=1}^{T} \prod_{i=1}^{N} P\left(Z_t^i | Pa\left(Z_t^i\right)\right). \tag{5}$$

DBN is often seen as a generalization of other developments in temporal reasoning, such as the Hidden Markov Model (HMM) and the Kalman Filter Model (KFM). These models can be expressed in a compact form and are popular because of their fast learning and rapid inference techniques. A network example of DBN is shown in Figure 1.

In Figure 1, (a) represents the initial distribution B_0, (b) represents the conditional distribution$B__\rightarrow$, and (c) represents a network segment formed by the initial network and the transfer network.

2.3. Parameter Learning for Dynamic Bayesian Networks. Based on the above DBN structure, it is necessary to learn the DBN network parameters, i.e., the conditional probability table reflecting the strength of the correlation between the network nodes from a large amount of data, including observation probabilities$P(Z_0)$, $P(Z_1)$, and $P(Z_2)$ and transfer probabilities $PZ_0^{t+1}|Z_0^t$, $PZ_1^{t+1}|Z_1^t$, and $PZ_2^{t+1}|Z_2^t$. The higher the similarity $\log^{(P(E|\theta))}$ between parameter θ in the conditional probability table in the DBN and the training data set E, the more realistic the results of parameter learning will be. In this paper, the method of maximum release probability is chosen to solve the conditional probability table parameter 1, and the following maximum release probability equation is constructed:

$$\log^{(P(E|\theta))} = \log \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{i=1}^{p_i} \theta_{ijk}^{n_{ijk}}, \tag{6}$$

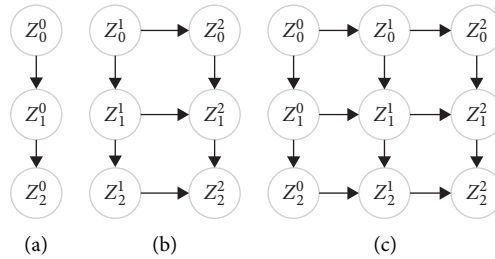


Figure 1: Example of a dynamic Bayesian network. (a) B_0: $P(Z_0)$. (b) $B_{\rightarrow}$: $P(Z_1|Z_2)$. (c) $B_0 \longrightarrow B_{\rightarrow}$: $(Z_0) \longrightarrow P(Z_1|Z_2)$

where i is all the nodes in the DBN, j is all the parents of Z_i, k is the state of Z_i, and n_{ijk} is the number of samples of the state k of the i-th node Z_i when its parent is the j-th combination:

$$\begin{cases} \text{Max} & \log^{P(E|\theta)} \\ \text{S.T.} & g_{ij}(\theta) = \sum_{k=1}^{r_i} \theta_{ijk} - 1 = 0. \end{cases} \tag{7}$$

The process of solving for the maximum probability of likelihood from equation (7) yields$\theta_{ijk} = (n_{ijk}/\sum_k n_{ijk})$.The sample data for parameter learning are preprocessed historical health status monitoring data, and experiments can be performed by some software such as GeNIe or MATLAB toolbox to complete the DBN parameter learning process.

2.4. DBN Inference Prediction and Evaluation. DBN inferential prediction and evaluation is a key step in the big data-based prediction and performance evaluation of DBN belt conveyor health status, and DBN inferential prediction and evaluation is a mathematical method to obtain a posterior distribution by updating the prior distribution of parameter values θ. After determining the DBN structure and parameter learning, it is necessary to analyse each variable in the network and make inferential predictions on the results of specific variables or events. In underground mine transportation system operation, when the belt conveyor health state changes, the monitoring data are entered into the DBN as evidence, and the dynamic Bayesian network could be updated to make inference prediction and evaluation. Finally, through inference calculations, the performance evaluation results of the equipment health state at the observation point at time t, and the prediction results of the trend of the belt conveyor health state from time t to time $t + h$ can be obtained.

Assuming that the group of observable nodes$Y = [y_{11}, \ldots, y_{1m}, \ldots, y_{T1}, \ldots, y_{Tm}]$ satisfy an independent identically distribution and the group of unknown network nodes are $Z = [z_{11}, \ldots, z_{1n}, \ldots, z_{T1}, \ldots, z_{Tn}]$, the dynamic Bayesian network inference rule for m known network nodes when a time slice contains n unknown network nodes is as follows:

$$P(Z|Y) = \sum_{y_{11}, y_{12}, \ldots, y_{Tm}} \frac{\prod_{ij} P\left(y_{ij}|Pa\left(Y_{ij}\right)\right) \prod_{ik} P\left(z_{ik}|Pa\left(Z_{ik}\right)\right) \prod_{ij} P\left(Y_{ijm} = y_{ij}\right)}{\sum_{z_{11}, \ldots, z_{1n}, \ldots, z_{T1}, \ldots, z_{Tn}} \prod_{ij} P\left(y_{ij}|Pa\left(Y_{ij}\right)\right) \prod_{ik} P\left(z_{ik}|Pa\left(Z_{ik}\right)\right)}, \tag{8}$$

where $P(z_{11}, \ldots, z_{1n}, \ldots, z_{T1}, \ldots, z_{Tn}|y_{11}, \ldots, y_{1m}, \ldots, y_{T1}, \ldots, y_{Tm})$ represents the conditional probability density of the variable $z_{11}, \ldots, z_{1n}, \ldots, z_{T1}, \ldots, z_{Tn}$ with respect to $y_{11}, \ldots, y_{1m}, \ldots, y_{T1}, \ldots, y_{Tm}$, i ranges from$i \in [1, T]$, j ranges from $j \in [1, m]$, k ranges from$k \in [1, n]$, Z_{ik} is the value of the unknown network node z_{ik}, $Y_{ij}$is the value of the observable nodey_{ij}, and$Pa(Y_{ij})$ is the parent node of y_{ij}.

The performance evaluation process is based on historical monitoring data to evaluate the current condition values, and the inference prediction process is based on historical monitoring data to predict the changes in the health status of the equipment. The DBN performance evaluation and inference prediction formulas are shown in the following equations:

$$P(Z_t|y_{1:\,t-1}) = \sum_{z_{t-1}} P(Z_t|z_{t-1}) P(z_{t-1}|y_{1:\,t-1}), \tag{9}$$

where $y_{1:\,t-1}$ is the evidence value of each condition indicator of belt conveyor health at moments 1 to $t - 1$ and Z_t is the probability of occurrence of belt conveyor health at moment t. $P(Z_t|y_{1:\,t-1})$ is the probability of evaluating the occurrence of belt conveyor health at moment t given that the evidence value of each health indicator at moments 1 to $t - 1$ is known:

$$P(Z_t|y_{1:\,t}) = P(y_t|Z_t)\left[\sum_{z_{t-1}} P(Z_t|z_{t-1}) P(z_{t-1}|y_{1:\,t-1})\right], \tag{10}$$

where $y_{1:\,t}$ is the evidence value for each of the health state indicators of the belt conveyor at moment t and $P(Z_t|z_{t-1})$ is the probability of inferentially predicting the occurrence of the health state of the belt conveyor from moment t to moment t, given that the evidence value for

each of the health state indicators from 1 to moment t is known.

2.5. Data-Driven DBN Prediction and Evaluation Methods. With the development of Internet of things in underground coal mine, more and more belt conveyor state data could be obtained by different kind sensor on it. Therefore, it is more feasible than ever before to precisely predict the health state and evaluate the performance of belt conveyor. As shown in Figure 2, the steps for data-driven DBN prediction and evaluation are as follows:

Step 1. Collect belt conveyor health status system monitoring data and preprocess to get a labelled feature data set

Step 2. Construct the DBN Topology

Step 3. Dynamic Bayesian network parameter learning

Step 4. Identifying observational evidence, dynamic Bayesian network evaluation, and inferential prediction

For underground mine belt conveyor health state prediction and performance evaluation, the above proposed method is mainly based on machine learning approaches. With the data obtained from the sensor or monitoring system, feature engineering is needed in the early stage where raw data should be processed and the key features should be selected to reduce the dimensionality of the training problem. Based on the DBN model, the performance evaluation and health state prediction could be conducted by inference.

3. Condition Prediction and Performance Evaluation of DBN Model

As a key equipment in coal mines, the actual performance of the belt conveyor will decrease over time due to various factors, leading to an increase in the probability of machine failure. To avoid economic losses and casualties due to sudden belt conveyor failure, it is necessary to provide enough time for maintenance, make a reasonable assessment of its current operating state, and make reasonable predictions of future changes in its operating state. A data-driven DBN belt conveyor health status prediction and performance evaluation method are proposed by combining real-time big data of belt conveyor health monitoring system and the advantages of deep learning. The flowchart of the method is shown in Figure 3, after which the specific steps model application are as follows:

(1) DBN network model construction: first, the belt conveyor health status historical data are processed, and the data are discretized to obtain the belt conveyor health status samples. Then, an initial DBN network model is constructed by using the experience of experts, and the training set data are parameterized to obtain a data-driven DBN belt conveyor health status prediction and performance evaluation model.

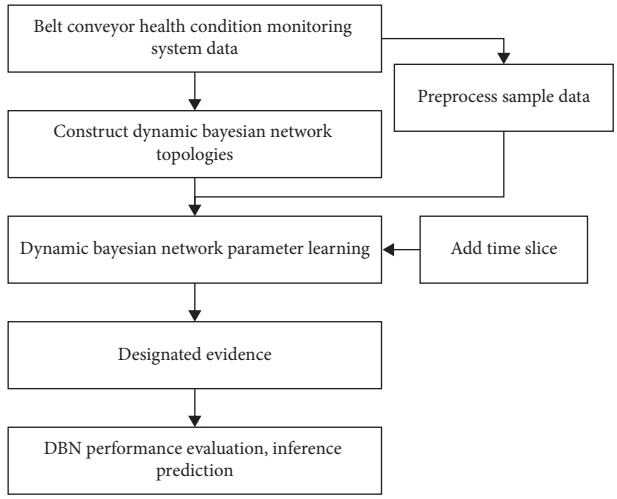


Figure 2: Dynamic bayesian network modeling flowchart.

(2) Belt conveyor health status prediction and performance evaluation: the values of belt conveyor health status parameters obtained from the belt conveyor health status monitoring system are entered into the DBN network model as evidence, and the DBN model is used to evaluate future belt conveyor health status, predict future belt conveyor health status changes, and propose safety maintenance decisions.

4. Case Study

4.1. Data Sources. The data used in this case are all from the actual data collected on-site by a new system of Internet of Things (IOT) in a coal mine in Shanxi province. From 2012, the coal mining company carried out a national demonstration project of the Internet of Things and introduced a new system of mine Internet of Things consisting of sensors, cloud platforms, and software systems. The enterprise has achieved certain results in data analysis after the development of two stages of the perceptual mine and the current development of intelligent mines. The big data provided by the mine IOTs are highly accessible and can realize real-time automatic collection of information, high-speed network transmission, standardized integration, three-dimensional visualization simulation, automatic operation, and intelligent decision-making. The real-time monitoring system of belt conveyor operation status can obtain eight types of typical belt conveyor health status, specifically involving belt stacking, belt fumes, belt overload, belt longitudinal tear, belt runaway, belt slip, belt breakage, normal system functions as shown in Figure 4. In this paper, the health status of a belt conveyor located in the main transport belt system of the mine's lean production platform is selected for the prediction and performance evaluation study.

The health status of a belt conveyor can be represented by a series of parameters that reflect its status. There are two ways to obtain health parameters, one is through the sensor in real time, and the other is from the periodic inspection statistics. The real-time monitoring data obtained from the sensors in the monitoring system include characteristic

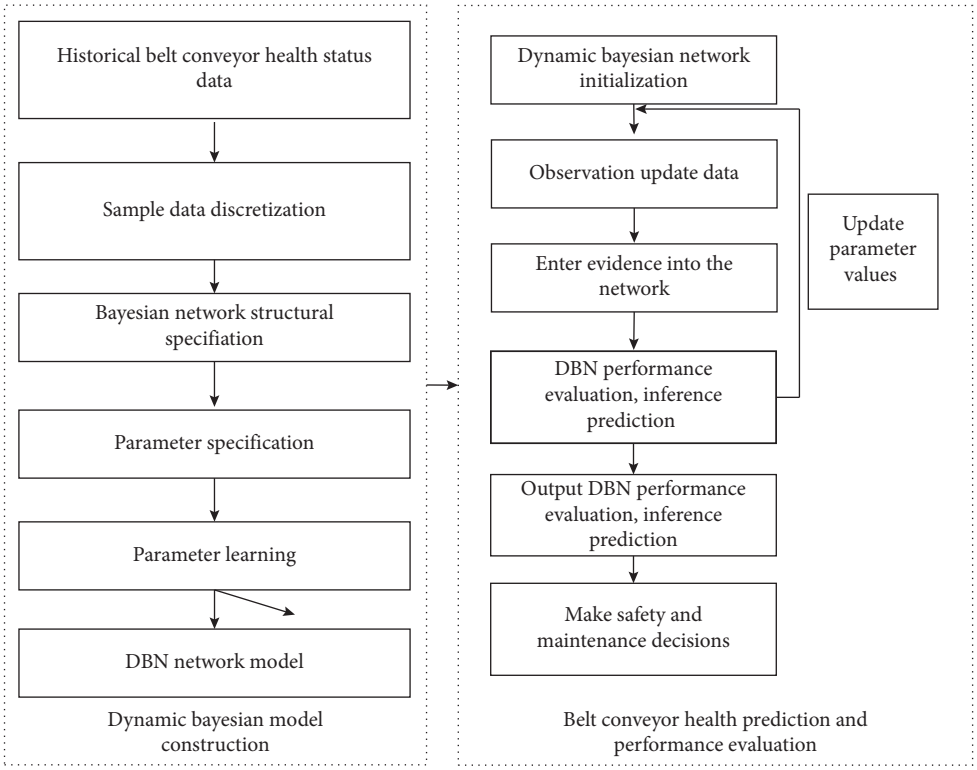


Figure 3: Mode application flow chart.

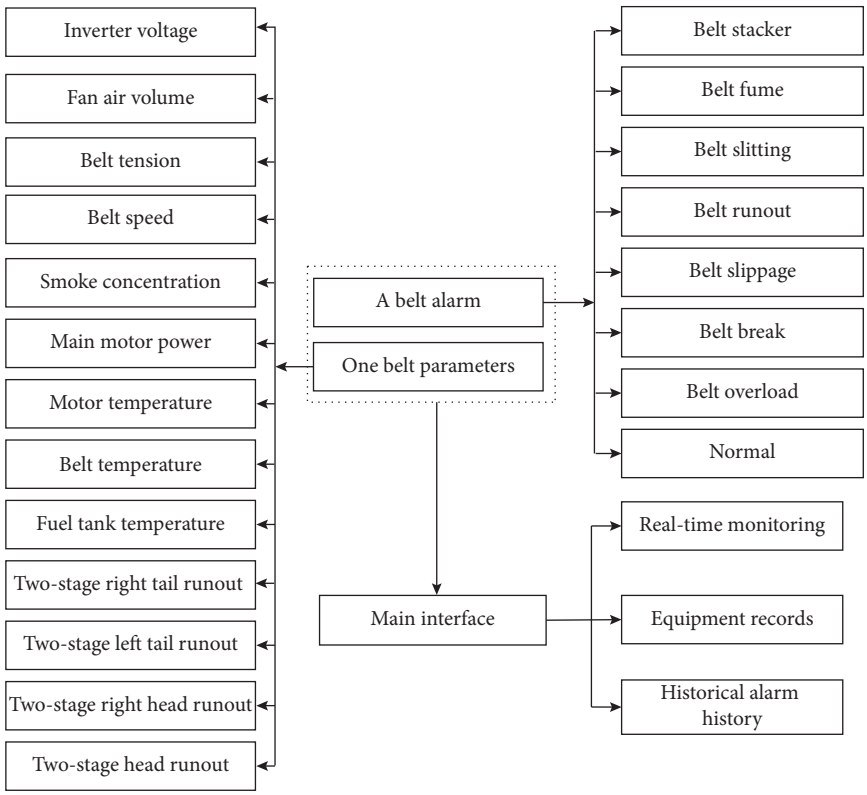


Figure 4: Belt conveyor health condition monitoring system functions.

parameters that can accurately reflect the operating status of the equipment and are easy to measure, specifically power, temperature, speed, offset, smoke concentration, conveyor belt tension, vibration, voltage, current, flow pressure, etc. The health parameters can be obtained from the sensors in real time, and the health parameters have different timeliness in predicting the health status. In this paper, only the health parameters collected by the coal mine belt conveyor operating status real-time monitoring system over a period are selected for analysis, and the sample data are shown in Table 1.

The above characteristic parameters constitute the set of variables of the dynamic Bayesian network and are denoted as $Z = \{Z_1, Z_2, \ldots, Z_{14}\}$,where$Z_1 \sim Z_{13}$represents main motor power, motor temperature, belt temperature, fuel tank temperature, two-stage right tail runout, two-stage left tail runout, two-stage right head runout, two-stage head runout, smoke concentration, belt speed, belt tension, and fan air volume, inverter voltage, andZ_{14} is the belt conveyor health, respectively. Status levels, corresponding to discretization values, represent the belt conveyor health status of very good, good, fair, poor, and very poor. Since the values of some of the characterization parameters are continuous, the variables are discretized as discontinuous. Based on expert experience and the statistical eigenvalues of the different variables, the continuous intervals are divided to obtain the discretization results shown in Table 2, where the corresponding discretization values are the risk level of each variable.

Due to the lack of sample data on the health status of belt conveyors, this paper collects the abnormal operation data from belt maintenance personnel and compiles the sample data containing abnormal operation data and real-time monitoring data of the normal operation of the system. The sampling frequency of the system real-time monitoring data was recorded every 10 seconds, and the recording frequency of manual statistical data was recorded once a week. The 17280 historical monitoring data within 48 hours were taken according to the time series, the time step of the dynamic Bayesian network was set to 2, and the first 17200 data and the last 80 data were selected as the training set of the model, respectively.

4.2. Belt Health Prediction and Performance Evaluation Model for Main Belt Conveyors

4.2.1. Determination of the DBN Structure and Parameter. After identifying the nodes of a dynamic Bayesian network and determining the node values, it is necessary to construct the relationships between the nodes. In this paper, the expert knowledge method is chosen to construct the node relationships, and the set of relationships is established sequentially as follows:

$$A = \{Z_{14} \longrightarrow Z_i\}, \tag{11}$$

where i takes the range of $i \in (1, 2, 3, \ldots, 13)$. The initial dynamic Bayesian network through the set of relations is shown in Figure 5.

After the initial dynamic Bayesian network structure is determined, the DBN parameters are learned from a sample of belt health data from the main conveyor section of the belt. The learning process is dynamic and the learned DBN parameters become more and more accurate as the operating time of the belt conveyor changes.

4.2.2. DBN Prediction and Performance Evaluation. The evidence and posterior probability distributions in the DBN network model are distributed according to time, and this paper selects the test set data in the data set to verify the contribution of the model in belt conveyor health status prediction and performance evaluation. First, we set up the time evidence and divide the test set of belt conveyor health monitoring data into 40 test set groups, and the eigenvalues at the moment $t = 0$ of each group are entered into the belt health prediction model to evaluate the performance of belt conveyor health in the first ten seconds and predict the belt conveyor health at the moment $t = 1$ as evidence. The evidence data of test set 1 is shown in Table 3.

Taking the test set 1 as an example, the belt conveyor health status data at $t = 0$ is input into the model as evidence assignment, and the belt conveyor health status evaluation at $t = 0$ and the belt conveyor health status prediction at $t = 1$ can be obtained, respectively, as shown in Figure 6. The probability distribution of each network node can be clearly seen from the bar graph of each node in the figure, and each health state has a different conditional probability table at different network nodes. The probability of belt conveyor health condition at $t = 0$ and $t = 1$ changes, and the probability of belt coal stacking increases significantly, and belt safety managers should take corresponding measures to manage the belt coal stacking phenomenon. Comparing the prediction and performance evaluation results of test set 1 with the actual situation, it can be found that there is inconsistency between the prediction data in test set 1 and the actual data, in which the prediction result of node X11 deviates greatly from the actual. The historical data of belt conveyor health condition and belt tension of belt conveyor has been in a relatively safe range for the past seven days, so it is judged that the excessive belt tension on that day is an emergency, while other inconsistent prediction results are in a small error range. The model is more accurate for belt conveyor health condition prediction and performance evaluation, which meets the basic requirements of actual belt conveyor health condition prediction and performance evaluation.

Set the experimental step to 10 to get the trend of the health state of the belt conveyor, as shown in Figure 7.

In Figure 7, “BCS”, “BS”, “LTFB”, “BD”, and “BSL”, “BB”, “BO”, and “N” represent the eight types of belt conveyor health, respectively. “BCS” is the belt stacker, “BS” is the belt fume, “LTFB” is the belt slitting, “BD” for the belt runout, “BSL” for the belt slip, “BB” for the belt breakage, and “BO” for the belt overload. “N” is the normal condition. The eight belt conveyor health states have changed over time, with the probability of a belt coal stacking condition increasing significantly. At this point, the belt safety manager

Table 1: Health characteristics of coal mine belt conveyors in operating condition.

Characterization	Belt stacker	Belt fume	Belt slitting	Belt runout	Belt slippage	Belt break	Belt overload	Normal
Main motor Power (KW)	345.23	345.65	284.51	283.29	206.45	286.9	348.25	248.55
Motor temperature (°C)	35.2	34.6	36.7	34.8	82.1	86.5	36.3	34.9
Belt temperature (°C)	55.8	55.5	56.5	55.7	88.3	86.5	55.6	55.6
Fuel tank temperature (°C)	56.4	55.6	56.2	56.1	55.9	58.9	56.3	55.9
Two-stage right tail runout (cm)	7	0	−53	5	0	−3	0	0
Two-stage left tail runout (cm)	−7	0	53	−5	0	3	0	0
Two-stage right head runout (cm)	0	0	−5	24	0	−63	13	0
Two-stage head runout (cm)	0	0	5	−24	0	63	−13	0
Smoke concentration (mg•m-2)	0.26	0.52	0.13	0.14	0.13	0.16	0.19	0.14
Belt speed (m•s-1)	2.45	3.51	2.55	3.53	3.51	0	2.62	3.51
Belt tension (kg)	82.3	82.1	86.5	87.4	77.1	89.6	92.4	80.0
Fan air volume (m•s-1)	6.15	6.19	6.13	6.12	6.13	6.15	6.15	6.14
Inverter voltage (v)	20.5	20.1	19.6	19.5	19.4	19.6	20.5	19.4

Table 2: Variable interval discretization classification.

Variables	Variable intervals	Corresponding discretized values
Main motor power (KW)	[0,248), [248, -]	1,2
Motor temperature (°C)	[0,70], [70,75], [75,80], [80, -]	1,2,3,4
Belt temperature(°C)	[0,55], [55,85], [85,90], [90, -]	1,2,3,4
Fuel tank temperature(°C)	[0,85], [85,90], [90,95], [95, -]	1,2,3,4
Two-stage right tail runout (cm)	0	1,2
Two-stage left tail runout (cm)	[1], [0]	1,2
Two-stage right head runout (cm)	[1], [0]	1,2
Two-stage head runout (cm)	[1], [0]	1,2
Smoke concentration ($mg{\cdot}m^{-2}$)	[1], [0]	1,2
Belt speed ($m{\cdot}s^{-}1$)	[0,3.15], [3.15,3.5], [3.5,4], [4, -]	1,2,3,4
Belt tension (kg)	[1], [0]	1,2
Fan air volume ($m{\cdot}s^{-}1$)	[0.25,6], [6,8,8,10,10,15]	1,2,3,4
Inverter voltage (v)	[0,20], [20,25], [25, -]	1,2,3
Health status levels	[0,0.2], [0.2,0.4], [0.4,0.6] [0.6,0.8], [0.8,1]	1,2,3,4,5

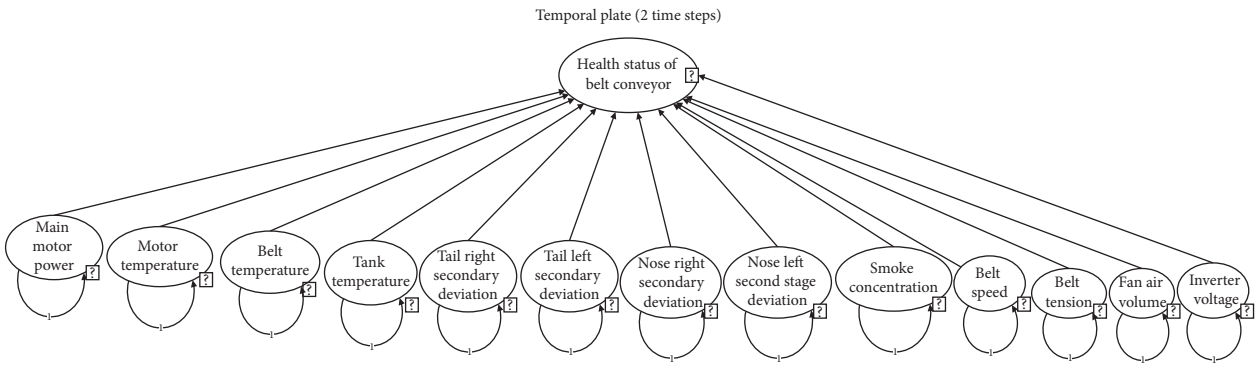


Figure 5: Initial dynamic bayesian network model.

is required to deal with the failure condition in a timely manner to prevent more serious damage.

4.3. Results and Discussion. With the development of IOT application in underground mine, more and more data are obtained. Meanwhile, more machine learning method could be used to fill the gap between data and useful information. To apply a dynamic Bayesian network-based model for equipment health prediction and performance evaluation, there are 2 issues need to be addressed:

(1) Compared with other prediction methods, DBN construction can combine both expert knowledge and machine equipment operating condition data to reduce model construction difficulty and use probability distribution tables and graphical structure to represent the uncertainty relationship between

Table 3: Test set 1 node evidence data table.

Characterization parameters	$t=0$ moment eigenvalues	$t=0$ moment corresponds to the value of the discretization	$t=1$ moment eigenvalue	$t=0$ moment corresponds to the value of the discretization
Main motor Power (kW)	283.29	2	345.23	2
Motor temperature (°C)	34.8	1	35.2	1
Belt temperature (°C)	55.7	2	55.8	2
Fuel tank temperature (°C)	56.1	1	56.4	1
Two-stage right tail runout (cm)	0	2	0	2
Two-stage left tail runout (cm)	0	2	0	2
Two-stage right head runout (cm)	1	1	0	2
Two-stage head runout (cm)	0	2	0	2
Smoke concentration ($mg \bullet m.s^{-1}$))	0	2	0	2
Belt speed ($m \bullet s^{-1}$)	3.53	2	2.46	2
Belt tension (kg)	87.5	2	104.2	2
Fan air volume ($m \bullet s^{-1}$)	6.12	2	6.16	2
Inverter voltage (v)	19.6	1	20.6	1

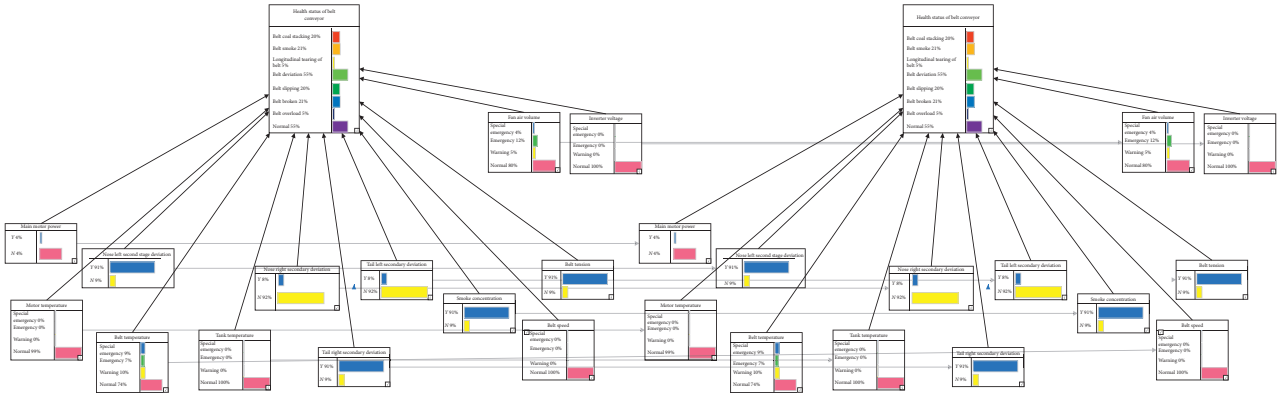


Figure 6: DBN prediction and performance evaluation results of belt conveyor.

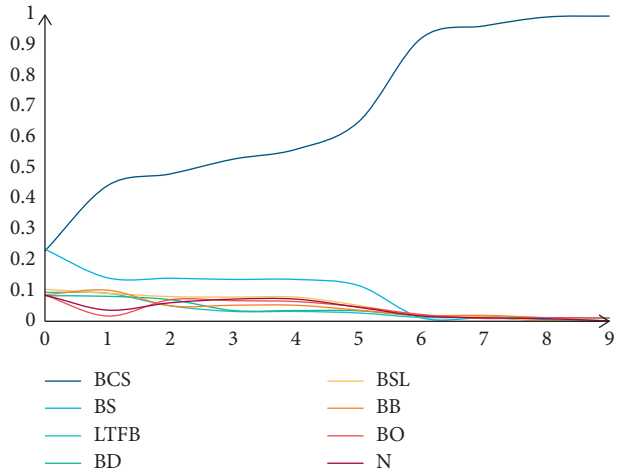


Figure 7: Trend of belt conveyor health status.

variables in a temporal context, which is basically in line with the need of mining uncertainty relationship between equipment health condition and characteristic parameters. At the same time, the prediction method can update the evidence using real-time updated data from machine equipment, thus reducing errors in prediction and performance evaluation.

(2) Among the 40 test sets, some of them have more than 5 deviations from the prediction, and the prediction effect is not so ideal, while the better-performing test sets have more conformity between the predicted and actual conditions. Analysing these test sets with unsatisfactory prediction results, the following possibilities can be suggested: ① The possibility of contingency: the possibility of contingency is mentioned in the analysis of the Z11 node in test set 1. This refers to a situation that has been stable for the current time, but the current state is less likely to occur. A contingency does not mean a situation that should not happen but rather a situation that happens with a small probability. This is a possibility of error because it is a common occurrence in real problems. ② Changes in influencing factors: since the data used in this paper are for a 48-hour period, the time span of the data is small, and during this period, there may be factors related to the improvement of belt conveyor maintenance technology, etc., which may have some impact on the health status. For example, if there is a defect in the main motor, the indicator is at a high safety level for the first 5 hours, but after the 5th hour, up to the 48th hour, the level of risk is often high, but in general, it has been significantly reduced and the trend of the risk level is stable. Therefore, this is a possible scenario for error generation.

5. Conclusions

The fault of belt conveyor in underground mine is analyzed and fault feature are selected from the monitoring Indicators. Based on data preprocessing, a dynamic Bayesian network is constructed after which the parameter is obtained by learning algorithm. It is proved that the proposed method is effective to assess and predicted the health performance of the conveyor belt. The main conclusions are as follows:

(1) The prediction and performance evaluation method proposed in this paper is based on a real-time data-driven dynamic Bayesian network model, and the accuracy of the results is 92% and 80% compared with the actual situation. The model could predict and evaluate the health status of belt conveyors and improve the accuracy of prediction and performance evaluation.

(2) Based on the analysis of the results, it is recommended that the data volume of the belt conveyor performance monitoring system should be increased to improve the prediction and warning efficiency of the belt conveyor from the source and avoid accidents caused by unstable belt health status.

(3) The belt conveyor health state prediction and performance evaluation are meaningful to promote the solution of coal mine machinery reliability problems and thus improve the reliability of the coal mine machinery system and ensure the normal implementation of coal mine safety management. In actual production, the mine machinery system is mostly a linked multidevice joint production system. Therefore, the method can be applied to the prediction of equipment health status and performance evaluation of multiple machines.

Data Availability

The data used to support the study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This study was supported by the National Key Research and Development Project of China (grant no.: 2017YFC0804408).

References

[1] C. J. Alonso-Gonzalez, N. Moya, and G. Biswas, "Dynamic Bayesian network factors from possible conflicts for continuous system diagnosis," *In Advances in Artificial Intelligence*, Springer, Berlin, Germany, 2011.

[2] W. Kawalec, R. Król, and N. Suchorab, "Regenerative belt conveyor versus haul truck-based transport: polish open-pit mines facing sustainable development challenges," *Sustainability*, vol. 12, no. 21, p. 9215, 2020.

[3] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[4] Y. G. Lei and Z. J. He, "Advances in applications of hybrid intelligent fault diagnosis and prognosis technique," *Journal of Vibration and Shock*, vol. 30, no. 9, pp. 129–135, 2011.

[5] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, "Prognostics and health management design for rotary machinery systems-reviews, methodology and applications," *Mechanical Systems and Signal Processing*, vol. 42, no. 1-2, pp. 314–334, 2014.

[6] M. Ma, C. Sun, and X. Chen, "Deep coupling autoencoder for fault diagnosis with multimodal sensory data," *IEEE Trans. Ind. Informat.*vol. 14, no. 3, pp. 1137–1145, 2018.

[7] C. Li, R.-V. Sanchez, G. Zurita, M. Cerrada, D. Cabrera, and R. E. Vásquez, "Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals," *Mechanical Systems and Signal Processing*, vol. 76-77, pp. 283–293, 2016.

[8] J. Gai and Y. Hu, "Research on fault diagnosis based on singular value decomposition and fuzzy neural network," *Shock and Vibration*, vol. 2018, Article ID 8218657, 7 pages.

[9] C. Sun, M. Ma, Z. Zhao, S. Tian, R. Yan, and X. Chen, "Deep transfer learning based on sparse autoencoder for remaining useful life prediction of tool in manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2416–2425, Apr. 2019.

[10] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213–237, 2019.

[11] C. F. Sun, Y. R. Wang, Y. Shen et al., "Fault diagnosis of planetary gearbox based on adaptive parameter variational mode decomposition," *Journal of Aerospace Power*, vol. 33, no. 11, pp. 205–214, 2018.

[12] M. Ma, C. Sun, X. Chen, X. Zhang, and R. Yan, "A deep coupled network for health state assessment of cutting tools based on fusion of multisensory signals," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6415–6424, Dec. 2019.

[13] X. Zhang, Z. Liu, Q. Miao, and L. Wang, "An optimized time varying filtering based empirical mode decomposition method with grey wolf optimizer for machinery fault diagnosis," *Journal of Sound and Vibration*, vol. 418, pp. 55–78, 2018.

[14] Y. G. Lei, F. Jia, X. Zhou et al., "A deep learning-based method for machinery health monitoring with big data," *Journal of Mechanical Engineering*, vol. 51, no. 21, pp. 55–62, 2015.

[15] Q. Jin, Y. R. Wang, and J. Wang, "Planetary gearbox fault diagnosis based on multiple feature extraction and information fusion combined with deep learning," *Journal of Mechanical Engineering*, vol. 30, no. 2, pp. 74–82, 2019.

[16] Y. Lei, F. Jia, J. Lin, S. Xing, and S. X. Ding, "An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 5, pp. 3137–3147, 2016.

[17] X. Li, W. Zhang, and Q. Ding, "Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction," *Reliability Engineering and System Safety*, vol. 182, pp. 208–218, 2019.

[18] J. Gai, J. Shen, H. Wang et al., "A parameter-optimized DBN using GOA and its application in fault diagnosis of gearbox," *Shock and Vibration*, vol. 2020, p. 11, Article ID 4294095, 2020.

[19] J. Hu, Z. L. Zhang, and Y. Wang, "An intelligent fault diagnosis system for process plant using a functional HAZOP and DBN integrated methodology," *Engineering Applications of Artificial Intelligence*, vol. 45, pp. 119–135, 2015.

[20] R. Liu, B. Yang, E. Zio, and X. Chen, "Artificial intelligence for fault diagnosis of rotating machinery: a review," *Mechanical Systems and Signal Processing*, vol. 108, pp. 33–47, 2018.

[21] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with convolutional bi-directional LSTM networks," *Sensors*, vol. 17, no. 2, 2017.

[22] X. Chen, Z. Shen, Z. He, C. Sun, and Z. Liu, "Remaining life prognostics of rolling bearing based on relative features and multivariable support vector machine," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 227, no. 12, pp. 2849–2860, 2013.

[23] E. Chemali, P. J. Kollmeyer, M. Preindl, R. Ahmed, and A. Emadi, "Long short-term memory networks for accurate state-of-charge estimation of li-ion batteries," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 8, pp. 6730–6739, 2018.

[24] S. Wang, X. Chen, I. W. Selesnick, Y. Guo, C. Tong, and X. Zhang, "Matching synchrosqueezing transform: a useful tool for characterizing signals with fast varying instantaneous frequency and application to machine fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 100, pp. 242–288, 2018.

[25] S. Yin, G. Wang, and H. R. Karimi, "Data-driven design of robust fault detection system for wind turbines," *Mechatronics*, vol. 24, no. 4, pp. 298–306, 2014.

[26] G. W. Vogl, B. A. Weiss, and M. Helu, "A review of diagnostic and prognostic capabilities and best practices for manufacturing," *Journal of Intelligent Manufacturing*, vol. 6, 2016.

[27] L. Bao, C. Juan, J. Li, and Y. Zhang, "Boosted near-miss undersampling on SVM ensembles for concept detection in large-scale imbalanced datasets," *Neurocomputing*, vol. 172, pp. 198–206, 2016.

[28] Y. Wu, B. Jiang, N. Lu, and Y. Zhou, "Bayesian network based fault prognosis via Bond graph modeling of high-speed railway traction device," *Mathematical Problems in Engineering*, vol. 2015, Article ID 321872, 2015.

Hindawi
Shock and Vibration
Volume 2021, Article ID 6627305, 8 pages
https://doi.org/10.1155/2021/6627305

Research Article

Multidomain Feature Fusion for Varying Speed Bearing Diagnosis Using Broad Learning System

Tingting Wu,[1] Yufen Zhuang,[1] Bi Fan,[1] Hainan Guo,[1] Wei Fan,[2] Cai Yi,[3] and Kangkang Xu[4]

[1]*College of Management, Shenzhen University, Shenzhen 518061, China*
[2]*School of Mechanical Engineering, Jiangsu University, Zhenjiang 212013, China*
[3]*State Key Laboratory of Traction Power, Southwest Jiaotong University, Chengdu 610031, China*
[4]*School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China*

Correspondence should be addressed to Bi Fan; fanbi@outlook.com

Received 18 December 2020; Revised 30 December 2020; Accepted 8 January 2021; Published 29 January 2021

Academic Editor: Tangbin Xia

Bearing is one of the most critical mechanical components in rotating machinery. To identify the running status of bearing effectively, a variety of possible fault vibration signals are recorded under multiple speeds. However, the acquired vibration signals have different characteristics under different speeds and environment interference, which may lead to different diagnosis results. In order to improve the fault diagnosis reliability, a multidomain feature fusion for varying speed bearing diagnosis using broad learning system is proposed. First, a multidomain feature fusion is adopted to realize the unified form of vibration characteristics at different speeds. Time-domain and frequency-domain features are extracted from the different speeds vibration signals. Then, the broad learning system is employed with the fused features for classification. Our experimental results suggest that, compared with other machine learning models, the proposed broad learning system model, which makes full use of the fused feature, can improve the diagnosis performance significantly in terms of both accuracy and robustness analysis.

1. Introduction

Rolling elements bearings are important machine elements that are widely used in railway wheels, wind turbines, gearboxes, pumps, and helicopter transmissions [1]. The running state of axle bearings, as the core component of high-speed trains, plays an important role in the safe and stable operation of the high-speed rail [2]. The railway axle bearings can support rotating machine elements and transfer loads of machine components under the fast-running state of a train. As the speed increases, the rolling bearings are operated under harsh conditions like heavy loads, long-term alternating stresses, and natural wear [3]. The axle bearings operating under these circumstances are prone to failures such as pitting, spalls, and axle burn-off [4]. If these bearing faults are ignored, this may lead to equipment damage and even cause serious safety accidents [5]. Thus, it is necessary to detect the bearing fault and give early warning before affecting daily operations. Effective fault diagnosis of bearings can prevent potential accidents and reduce unexpected economic loss [4]. It has been found that bearing faults are often accompanied by abnormal vibration signals, such as outer ring inner surface faults, cage faults, roller faults, and the compound faults [6]. Therefore, analyzing and mining diagnostic information from vibration data have important scientific significance and high application value.

At present, it is impossible to identify the faults from the vibration signal directly. The vibration signal is vulnerable to the operating environment and unknown factors, such as the track impact, speed change, and the vehicle body vibration [7]. It is necessary to eliminate the interference and extract useful information that can reflect the health of the bearing. Feature extraction is a key step to solve this problem. It can transform the original vibration signals onto the statistical parameters reflecting diverse symptoms of bearing defects [8]. Many methods have been developed, such as Fourier

transform (FT), short-time Fourier transform (STFT), wavelet transform (WT), and wavelet packet decomposition (WPD) [9–15]. Bouzida et al. [16] employed discrete wavelet transform to obtain information about the health state of a system from stator signal over a wide range of frequencies. Li et al. [17] proposed a feature extraction and evaluation method based on statistical features. Zhang et al. [18] proposed an ant colony algorithm for synchronous feature selection in intelligent fault diagnosis of rotating machinery. Cong et al. [19] used the slip matrix construction method to extract features. However, the characteristics of vibration signal of bearing under different rotating speeds are rarely discussed. Some researchers have developed several fault diagnosis methods according to different speeds, respectively [2, 20]. This method seems to work well dealing with different speeds, but it lacks convenience in practice. In addition, the characteristics of the vibration signal across different speeds have not been studied.

The vibration signal can be regarded as an expression of a moving process in time domain and frequency domain [3]. Operating speed of the bearings will also impact the moving process [21]. Therefore, the inherent characteristic of vibration signal of variable speed should be extracted from the time and frequency domain together to achieve better fault diagnosis performance [22]. Thus, a unified description of multidomain fusion for varying speed should be studied [21]. The accuracy of detection mainly depends on the quality of features extracted from the vibration data. Therefore, a multidomain features fusion is proposed to present the inherent characteristics of the vibration signal for varying speed comprehensively. The common time and frequency features [20] involve mean value, variance, maximum, root mean square, etc. They can capture intrinsic information about bearing defects. For example, mean value in time domain represents central trend, and the variance in frequency domain conveys message of signal changes. We chose STFT to extract time domain features for its compatible implementation in the fast Fourier transform. Meanwhile, WPD was one of the most widely used and advanced technologies to analyze signal data in frequency domain. These two methods were combined to extract time and frequency features.

In recent years, machine learning algorithms, such as broad learning system (BLS), artificial neural network (ANN), extreme learning machine (ELM), support vector machine (SVM), and logistic regression (LR) have been applied successfully in fault diagnosis of roller bearings [23–26]. Sobie et al. [21] proposed a novel application of dynamic time warping (DTW) to bearing fault classification. Toma et al. [27] used KNN, decision tree, and random forest to evaluate the bearing faults. Zhang et al. [28] proposed a novel hybrid model using permutation entropy (PE), ensemble empirical mode decomposition (EEMD), and support vector machine (SVM) to detect roller bearing faults. Recently, BLS has received a lot of attention due to its outstanding performance in fault diagnosis. Zhao et al. [2] employed BLS as classifier to detect the bearing faults. Zhao et al. [29] proposed semisupervised broad learning system for fault diagnosis. However, these studies either directly use BLS without considering the inherent characteristics of the data or only consider the frequency-domain characteristics of the signal. It seems that the present work cannot make full use of BLS in fault diagnosis.

In this paper, we propose a multidomain feature fusion for varying speed bearing diagnosis using BLS. The diagnostic power of the method is attributable to two features: First, we extract the intrinsic vibration characteristics at multiple speeds. A multidomain fusion is adopted to realize the unified form of vibration characteristics; second, the kernel-based broad learning system has short computing time and good generalization ability. Different from previous studies, the original vibration data at different speeds are converted into unified time-domain and frequency-domain data. As far as we know, few studies have adopted this method to extract diagnostic information from different speeds at one time. Obviously, with multidomain fusion, the vibration data of different speeds can be explored thoroughly under a unified framework to obtain more dynamic fault information. The experiment results illustrate that the proposed method is significantly superior to some other machine learning models. Meanwhile, the diagnostic power of BLS with varying speed data is stronger than that with one speed, which may provide solid evidence that varying speed data are of great significance to fault diagnosis.

The remainder of this paper is organized as follows. Section 2 introduces the framework of the proposed diagnosis method. The proposed multidomain feature fusion and BLS are provided in Section 2. The empirical study is given in Section 3. Finally, Section 4 offers concluding work and implications for further research.

2. Methodology

2.1. Analytical Framework. The general scheme of applying the multidomain feature fusion for varying speed bearing diagnosis is shown in Figure 1. The proposed method includes three major steps:

(1) *Obtain the Vibration Signals under Different Rotating Speeds*. The raw data were collected from the sensors and segmented into training dataset and testing dataset.

(2) *Multidomain Feature Fusion*. The acquired vibration data were preprocessed with time-domain and frequency-domain transformation. The time statistical characteristics and the frequency spectra of the vibration signals provide potentially valuable fault information.

(3) *Fault Diagnosis Using BLS*. The obtained fused features were utilized to train the BLS and other machine learning classification models. Then the testing dataset is used to validate their performance.

2.2. Multidomain Feature Fusion for Varying Speed. Feature extraction plays an important role in fault diagnosis. In a condition monitoring system, the bearing is often running under different speeds. These vibration signals of varying speed contain a variety of possible fault information. In particular, the dynamic characteristics of different speeds

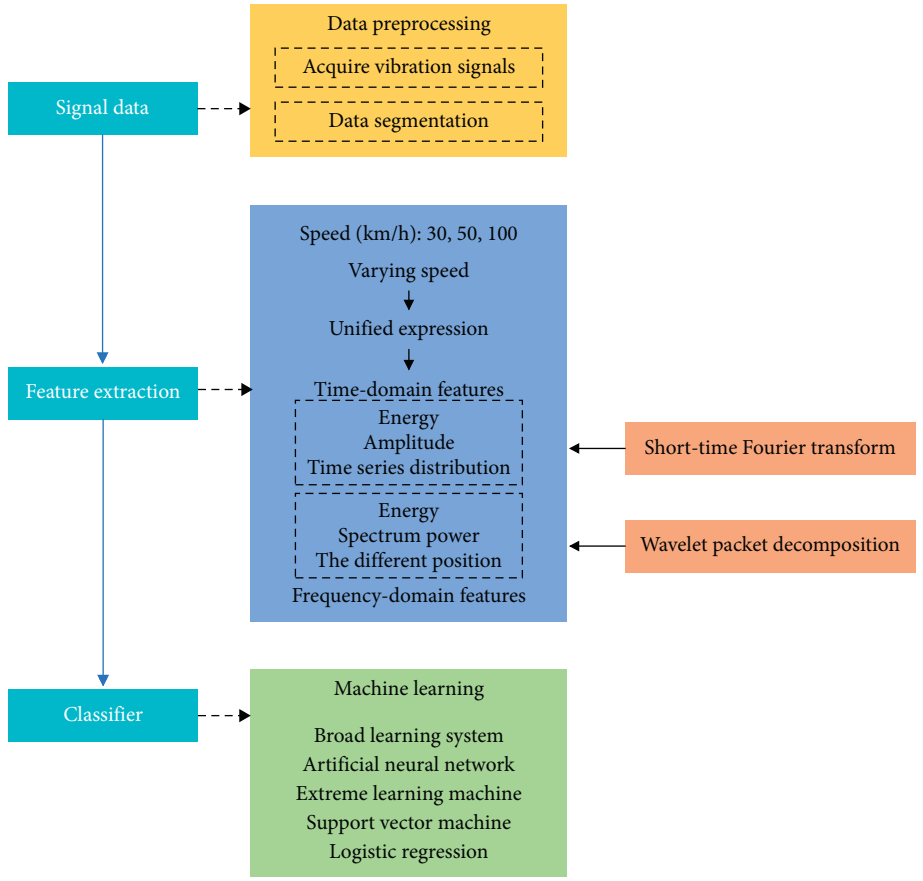


Figure 1: The framework of the proposed method.

should be considered together. The purpose of multidomain fusion is to produce more reliable and more accurate information from varying speed.

In order to capture the diagnostic information, a total of 50 features are extracted from each speed vibration signal, 18 time-domain features, and 32 frequency-domain features as listed in Tables 1 and 2, respectively. Table 1 presents the description of 18 time-domain features, including mean value, mean square, peak-to-peak value, impulse factor, and crest factor. Table 2 presents the definition of 32 frequency-domain features, including frequency center and RMS frequency. Among them, p_1 and $p_{16} - p_{18}$ denote the amplitude and energy of time-domain features. p_7 and $p_{11} - p_{12}$ reflect the time series distribution. p_{19} is the energy of frequency-domain features. $p_{21} - p_{24}$, p_{26}, and $p_{30} - p_{33}$ are the spectrum power. p_{25} and $p_{27} - p_{29}$ show the different positions of the main frequencies. The remaining features are extracted by wavelet packet analysis and expressed as equations $p_{34} - p_{50}$.

2.3. Broad Learning System. Broad learning system (BLS) was proposed by Professor Chen [30]. It is an incremental learning algorithm based on the random vector function link neural network (RVFLNN). For this method, firstly, the original data are projected in the feature space using a linear function and transformed into features notes of the BLS. Then enhancement notes are generated by feature notes randomly through a nonlinear activation function. Next, all feature nodes and enhancement nodes are directly connected to the output coefficients, which can be obtained by the ridge regression. Finally, the structure of BLS is constructed after the output weight is obtained. Compared with other deep networks, the structure of BLS is simple. The BLS establishes a flat network structure consisting of feature nodes, enhancement nodes, and output coefficient matrix. It can quickly extract features from new data and reduce retraining time. Therefore, the flat network of BLS can be more effective and efficient in classification and regression problems without deep architecture [29]. The BLS structure is shown in Figure 2 [30].

(1) For the input data X, the linear transformation function mapping is used to project data, which become the ith mapped features Z_i.

$$Z_i = \varnothing_i(XW_{ei} + \beta_{ei}), \tag{1}$$

where W_{ei} and β_{ei} are the random weights with the proper dimensions. Denote $Z_i \equiv [Z_1, \ldots, Z_n]$, which is the concatenation of all the first i groups of mapping features and n is the number of groups of feature nodes; $i = 1, \ldots, n$.

Table 1: Time-domain features.

$p_1 = (1/N)\sum_{i=1}^{N} x_i$	$p_2 = \max(x_i)$	$p_3 = \min(x_i)$
$p_4 = (1/N)\sum_{i=1}^{N} (x_i - p_1)^2$	$p_5 = \sqrt{p_4}$	$p_6 = ((1/N)\sum_{i=1}^{N} x_i^2)^{1/2}$
$p_7 = p_6/(1/N)\sum_{i=1}^{N} \lvert x_i\rvert$	$p_8 = (1/N)\sum_{i=1}^{N} (x_i - p_1/p_5)^4$	$p_9 = (1/N)\sum_{i=1}^{N} (x_i - p_1/p_5)^3$
$p_{10} = p_2 - p_3$	$p_{11} = \max(\lvert x_i\rvert)/p_6$	$p_{12} = \max(\lvert x_i\rvert)/(1/N)\sum_{i=1}^{N} \lvert x_i\rvert$
$p_{13} = p_2/p_7$	$p_{14} = p_2/p_6$	$p_{15} = p_8/p_6^4$
$p_{16} = (1/N)\sum_{i=1}^{N} \lvert x_i\rvert$	$p_{17} = \max(\lvert x_i\rvert)$	$p_{18} = ((1/N)\sum_{i=1}^{N} \sqrt{\lvert x_i\rvert})^2$

Note: x_i denotes a signal data for $i = 1, 2, \ldots, N$. N indicates the number of signal points.

Table 2: Frequency-domain features.

$p_{19} = (1/K)\sum_{i=1}^{K} s_i$	$p_{20} = ((1/K)\sum_{i=1}^{K} S_i^2)^{1/2}$	$p_{21} = (1/K-1)\sum_{i=1}^{K} (s_i - p_{19})^2$
$p_{22} = \sqrt{p_{21}}$	$p_{23} = \sum_{i=1}^{K} (s_i - p_{19})^3/K(p_{22})^3$	$p_{24} = \sum_{i=1}^{K} (s_i - p_{19})^4/K(p_{22})^2$
$p_{25} = \sum_{i=1}^{K} f_i s_i/\sum_{i=1}^{K} s_i$	$p_{26} = (\sum_{i=1}^{K} (f_i - p_{25})^2 s_i/K)^{1/2}$	$p_{27} = (\sum_{i=1}^{K} f_i^2 s_i/\sum_{i=1}^{K} s_i)^{1/2}$
$p_{28} = (\sum_{i=1}^{K} f_i^4 s_i/\sum_{i=1}^{K} f_i^2 s_i)^{1/2}$	$p_{29} = \sum_{i=1}^{K} f_i^2 s_i/(\sum_{i=1}^{K} s_i \sum_{i=1}^{K} f_i^4 s_i)^{1/2}$	$p_{30} = p_{26}/p_{25}$
$p_{31} = \sum_{i=1}^{K} (f_i - p_{25})^3 s_i/K(p_{26})^3$	$p_{32} = \sum_{i=1}^{K} (f_i - p_{25})^4 s_i/K(p_{26})^4$	$p_{33} = \sum_{i=1}^{K} (f_i - p_{15})^{1/2} s_i/K(p_{26})^{1/2}$
$p_{34} - p_{49} = E_i/\sum_{i=1}^{16}\sum E_i$	$p_{50} = -\sum_{i=1}^{16} E_i \log_2 E_i$	

Note: s_i shows a spectrum for $i = 1, 2, \ldots, K$. K describes the number of spectrum lines. f_i is the frequency value of the kth spectrum line.

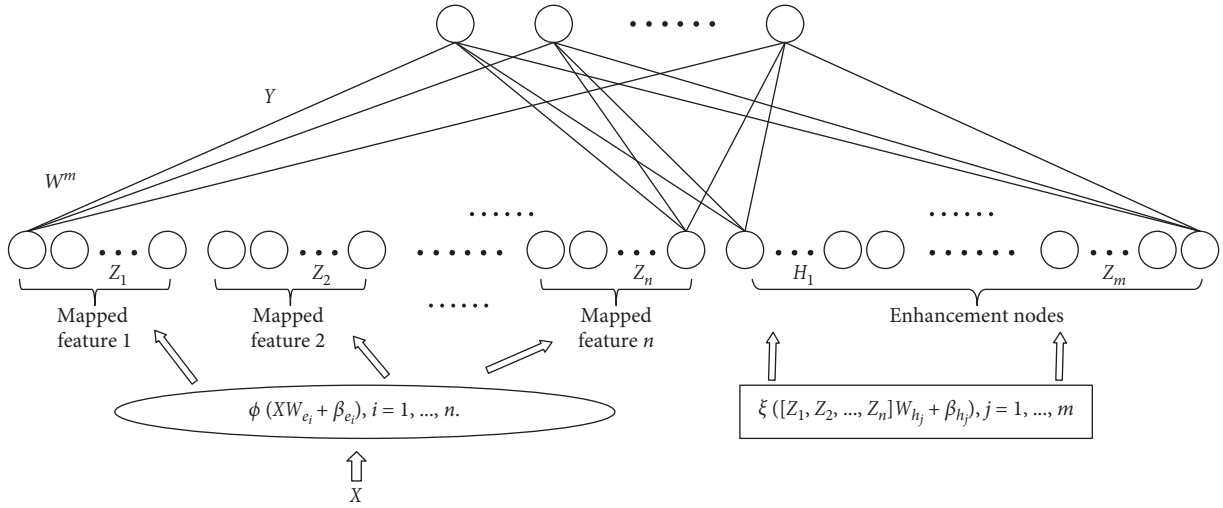


Figure 2: The BLS structure.

(2) Similarly, the jth group of enhancement nodes H_j can be generated using the nonlinear function transformation

$$H_j = \xi_i\left(Z_n W_{hj} + \beta_{hj}\right), \tag{2}$$

where W_{hj} and β_{hj} are random weight coefficients with appropriate dimensions. The concatenation of all the first j groups of enhancement nodes is denoted as $H^m \equiv [H_1, \ldots, H_m]$. m is the total number of enhancement nodes; $j = 1, \ldots, m$.

Hence, the BLS can be represented as the equation of the form

$$\begin{aligned} Y &= \left[Z_1, \ldots, Z_n \mid \xi\left(Z_n W_{h_1} + \beta_{h_1}\right), \ldots, \xi\left(Z_n W_{h_m} + \beta_{h_m}\right)\right] W^m \\ &= [Z_1, \ldots, Z_n \mid H_1, \ldots, H_m] W^m \\ &= [Z^n \mid H^m] W^m, \end{aligned} \tag{3}$$

where $W^m = [Z^n|H^m]^+ Y$. W^m are the connecting weights coefficients and can be easily computed using the ridge regression.

3. Experiment and Analysis

3.1. Experiment Data and Environment. In order to verify the effectiveness of the proposed method, the experiments were carried out on a rotary machine experimental platform, as shown in Figure 3. The platform can simulate various operating conditions of trains with infinite long rail through double-wheel reverse scrolling. The test rig can simulate various operating conditions of trains. The fault states of train can be imitated through artificially seeded defects in bench experiments. Experiments were designed with three faulty bearings and one normal bearing. Under each type of fault defects, the bearing was running at the speeds of 30 km/h, 50 km/h, and 100 km/h for 2 minutes, respectively, and the 10 seconds' vibration data were collected with the sampling frequency of 10 kHz. For each bearing defect in certain speed, 10,000 pieces of data were sampled. The fault types and speeds are shown in Table 3. According to the sampling frequency and rotating speed, the number of sampling points included in a cycle during the bearing rotation is calculated. Therefore, the original data were

Figure 3: Experimental setup.

Table 3: Description of bearing dataset.

Name	Fault type	Speed (km/h)	Fault classification
ZC_1	Outer race defects	30, 50, 100	1
ZC_2	Cage defect	30, 50, 100	2
ZC_3	Pin roller defect	30, 50, 100	3
ZC_4	Normal	30, 50, 100	4

Table 4: Comparison of classification accuracy.

Models	ZC_1	ZC_2	ZC_3	ZC_4	Average accuracy (%)
BLS	96.23	97.60	94.49	98.74	96.2
ANN	87.28	83.17	86.59	89.68	86.68
ELM	73.47	92.15	84.21	88.92	84.69
SVM	92.99	94.34	97.97	88.86	93.54
LR	74.13	88.65	82.82	89.32	83.73

divided into smaller datasets according to different faults and speeds.

In order to capture the dynamic diagnostic information, there are 18 time-domain features and 32 frequency-domain features extracted from each vibration signal, as listed in Tables 1 and 2. These features are fused to form the multidomain features. The obtained fused features can be utilized to train the BLS for fault diagnosis.

3.2. Fault Classification. To further verify the effectiveness of the proposed method, the multidomain fusion features are used to train the BLS, ANN, SVM, ELM, and LR as the classification model. The SVM classifier is a margin-based supervised machine learning method. SVM model can be effectively applied in nonlinearly separable data. The radial basis function was used in SVM model. ELM is a single hidden layer neural network algorithm and has been widely used in many fields because of simple mathematical description, lower computational burden, and faster learning speed [31]. ANN was inspired by biological nervous systems function of the man brain. It is suitable for complex natural systems by establishing relationships among highly anomalous nonlinear variables and producing sophisticated, accurate, and reliable results [32]. In this study, the number of hidden neurons of ELM and ANN models was determined by trial-and-error testing. LR is capable of bearing fault diagnosis for its high accuracy in the nearly linearly separable data.

The accuracy of each fault is used to evaluate the performance of these classifiers. The fault diagnosis performances of these models are shown in Table 4. The comparison results of accuracy of each fault and average accuracy are provided. The proposed BLS model has the highest average accuracy in testing compared to other machine learning methods. Furthermore, BLS constantly outperforms other methods for each fault, followed by SVM, ANN, and ELM, whereas LR ranks last.

The robustness of the proposed method is also assessed. Due to the random variables in BLS model, we ran the experiment ten times to obtain the average performance of BLS and analyzed its robustness according to standard deviation of accuracy. The results are provided in Figure 4. The standard deviations of accuracy are as follows: 0.009 (ZC_1), 0.007 (ZC_2), 0.012 (ZC_3), 0.005 (ZC_4), and 0.002 (Average). Generally, the performance of the BLS is stable, since the standard deviations from the accuracy of each fault are small. The standard deviation from average accuracy is smaller than others.

To further examine the effectiveness of the multidomain fusion for varying speed, the training data are classified into three types: "30 km/h," "50 km/h," and "100 km/h." These are vibration data under different speeds. When one dataset is used to train the BLS, the other two datasets are tested. For

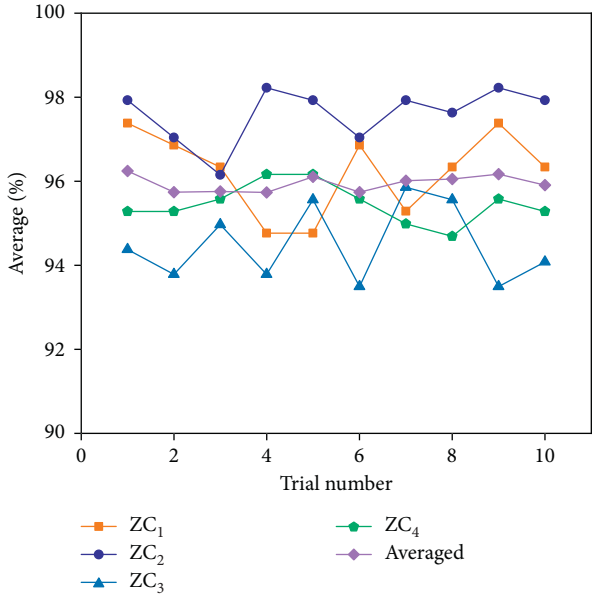


Figure 4: Results of 10 trials using BLS.

Table 5: Classification comparison of different combinations.

Models	Training dataset (km/h)	Testing dataset (km/h)	Average accuracy (%)
BLS	30	50	94.12
	30	100	79.24
	50	30	98.03
	50	100	85.63
	100	30	74.94
	100	50	91.32
ANN	30	50	87.33
	30	100	72.44
	50	30	80.02
	50	100	71.16
	100	30	79.46
	100	50	90.68
ELM	30	50	82.64
	30	100	68.65
	50	30	95.25
	50	100	81.46
	100	30	78.12
	100	50	70.18
SVM	30	50	78.91
	30	100	80.14
	50	30	94.33
	50	100	94.42
	100	30	80.95
	100	50	91.70
LR	30	50	83.05
	30	100	64.28
	50	30	95.51
	50	100	87.98
	100	30	63.35
	100	50	83.37

comparison with the BLS, we conducted other experiments using SVM, ANN, ELM, and LR to test the effectiveness of different combinations. The experimental results are recorded in Table 5. Generally, the performance of the five models with different combinations is not good. BLS is more accurate, followed by SVM, ANN, LR, and ELM. When the

speed of training dataset is higher than that of the testing dataset, the accuracy of the model seems better. For example, when the training dataset is "50 km/h" and testing dataset is "30 km/h," the classification accuracy of five models is good. In general, we can get some evidence that (1) there are declines in diagnosis accuracy; (2) the training and testing data with a large speed difference make the diagnosis more difficult; (3) different speeds contain different dynamic characteristics for fault diagnosis. It has been revealed that the classifier developed on one speed data may not be capable of good diagnosis on other speed data On the contrary, the method proposed in this paper has good performance of fault diagnosis under multispeed with one classifier.

4. Conclusion

In this paper, we proposed a diagnosis framework using multidomain feature fusion and machine learning to detect faults from vibration data at different speeds. The study considers the dynamic characteristics of multiple speeds together to obtain more comprehensive fault diagnosis information. Features extracted from time and frequency domains are combined into one vector to present the characteristics of multiple speeds. Then BLS is developed on these fused features for fault diagnosis. The experimental results suggest that the proposed BLS models with multidomain features can significantly improve the forecasting performance compared with other machine learning methods. Beside the accuracy, the robustness of the proposed BLS is also analyzed. Moreover, it is also verified that the classifier developed on one speed data may not be able to perform well on other speed data. However, this study does have some limitations, mainly because we only solved this problem at a certain level. There are still some problems to be investigated. The relationship between the vibration data of different speeds is still unknown. How to develop a diagnostic model for all speeds is also a problem Furthermore, the model could deal with multiple domain data such as multiple sensors and speeds. These questions should be studied in the future.

Data Availability

Data are available upon request to the corresponding author by e-mail.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. B. Randall, *Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications*, John Wiley & Sons, Hoboken, NJ, USA, 2011.

[2] H. Zhao, J. Zheng, J. Xu, and W. Deng, "Fault diagnosis method based on principal component analysis and broad learning system," *IEEE Access*, vol. 7, pp. 99263–99272, 2019.

[3] C. Zheng, D. Wang, W. Fan, K.-L. Tsui, and J. Lin, "EEMD-based steady-state indexes and their applications to condition monitoring and fault diagnosis of railway axle bearings," *Sensors*, vol. 18, no. 3, p. 704, 2018.

[4] C. Yi, J. Lin, W. Zhang, and J. Ding, "Faults diagnostics of railway axle bearings based on IMF's confidence index algorithm for ensemble EMD," *Sensors*, vol. 15, no. 5, pp. 10991–11011, 2015.

[5] A. Heng, A. C. C. Tan, J. Mathew, N. Montgomery, D. Banjevic, and A. K. S. Jardine, "Intelligent condition-based prediction of machinery reliability," *Mechanical Systems and Signal Processing*, vol. 23, no. 5, pp. 1600–1614, 2009.

[6] S. Tan, H. A. Toliyat, and X. Li, "Condition monitoring and fault diagnosis of electrical motors-a review," *IEEE Transactions on Energy Conversion*, vol. 20, no. 4, pp. 719–729, 2005.

[7] J. R. Stack, T. G. Habetler, and R. G. Harley, "Effects of machine speed on the development and detection of rolling element bearing faults," *IEEE Power Electronics Letters*, vol. 1, no. 1, pp. 19–21, 2003.

[8] M. Kang, J. Kim, J.-M. Kim, A. C. C. Tan, E. Y. Kim, and B.-K. Choi, "Reliable fault diagnosis for low-speed bearings using individually trained support vector machines with kernel discriminative feature analysis," *IEEE Transactions on Power Electronics*, vol. 30, no. 5, pp. 2786–2797, 2015.

[9] Y. Li, H. Zhao, W. Fan, and C. Shen, "Extended noise resistant correlation method for period estimation of pseudo-periodic signals," *IEEE Transactions on Instrumentation and Measurement*, pp. 1–1, 2021.

[10] W. Fan, Q. Zhou, J. Li, and Z. Zhu, "A wavelet-based statistical approach for monitoring and diagnosis of compound faults with application to rolling bearings," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1563–1572, 2017.

[11] Y. Fan, L. Li, L. Chen, and Z. Xu, "Bearing Fault Detection Via B-spline Constructed Sparse Method," *IEEE Transactions on Instrumentation and Measurement*, pp. 1–1, 2021.

[12] G. Chen, Q.-Y. Li, D.-Q. Li, Z.-Y. Wu, and Y. Liu, "Main frequency band of blast vibration signal based on wavelet packet transform," *Applied Mathematical Modelling*, vol. 74, pp. 569–585, 2019.

[13] D. Li, "Theory of communication. Part 1: the analysis of information," *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, 1946.

[14] H. Gao, L. Liang, X. Chen, and G. Xu, "Feature extraction and recognition for rolling element bearing fault utilizing short-time Fourier transform and non-negative matrix factorization," *Chinese Journal of Mechanical Engineering*, vol. 28, no. 1, pp. 96–105, 2015.

[15] Y. Liang, G. Xu, L. Liang, and K. Jiang, "Detection of weak transient signals based on wavelet packet transform and manifold learning for rolling element bearing fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 54-55, pp. 259–276, 2015.

[16] A. Bouzida, O. Touhami, R. Ibtiouen, A. Belouchrani, M. Fadel, and A. Rezzoug, "Fault diagnosis in industrial induction machines through discrete wavelet transform," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 9, pp. 4385–4395, 2010.

[17] W. Li, Z. Zhu, F. Jiang, G. Zhou, and G. Chen, "Fault diagnosis of rotating machinery with a novel statistical feature extraction and evaluation method," *Mechanical Systems and Signal Processing*, vol. 50-51, pp. 414–426, 2015.

[18] X. Zhu, W. Chen, B. Wang, and X. Chen, "Intelligent fault diagnosis of rotating machinery using support vector machine with ant colony algorithm for synchronous feature selection

and parameter optimization," *Neurocomputing*, vol. 167, pp. 260–279, 2015.

[19] F. Chen, W. Zhong, S. Tong, N. Tang, and J. Chen, "Research of singular value decomposition based on slip matrix for rolling bearing fault diagnosis," *Journal of Sound and Vibration*, vol. 344, pp. 447–463, 2015.

[20] Z. Chow and W. Li, "Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network," *IEEE Transactions on Instrumentation and Measurement*, vol. 99, pp. 1–10, 2017.

[21] C. Zhong, C. Freitas, and M. Nicolai, "Simulation-driven machine learning: bearing fault classification," *Mechanical Systems and Signal Processing*, vol. 99, pp. 403–419, 2018.

[22] B. Li, M.-Y. Chow, Y. Tipsuwan, and J. C. Hung, "Neural-network-based motor rolling bearing fault diagnosis," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 5, pp. 1060–1069, 2000.

[23] J. B. Ali, "Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals," *Applied Acoustics*, vol. 89, no. 3, pp. 16–27, 2015.

[24] Y. Tian, J. Ma, C. Lu, and Z. Wang, "Rolling bearing fault diagnosis under variable conditions using LMD-SVD and extreme learning machine," *Mechanism and Machine Theory*, vol. 90, pp. 175–186, 2015.

[25] K. Ma, X. Song, and D. Xue, "A roller bearing fault diagnosis method based on hierarchical entropy and support vector machine with particle swarm optimization algorithm," *Measurement*, vol. 47, pp. 669–675, 2014.

[26] D. H. Pandya, S. H. Upadhyay, and S. P. Harsha, "Fault diagnosis of rolling element bearing by using multinomial logistic regression and wavelet packet transform," *Soft Computing*, vol. 18, no. 2, pp. 255–266, 2014.

[27] R. N. Toma, A. E. Prosvirin, and J.-M. Kim, "bearing fault diagnosis of induction motors using a genetic algorithm and machine learning classifiers," *Sensors*, vol. 20, no. 7, p. 1884, 2020.

[28] X. Xhang, Y. Liang, and J. Zhou, "A novel bearing fault diagnosis model integrated permutation entropy, ensemble empirical mode decomposition and optimized SVM," *Measurement*, vol. 69, pp. 164–179, 2015.

[29] H. Zhao, J. Zheng, W. Deng, and Y. Song, "Semi-supervised broad learning system based on manifold regularization and broad network," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 3, pp. 983–994, 2020.

[30] C. L. P. Chen and Z. Liu, "Broad learning system: an effective and efficient incremental learning system without the need for deep architecture," *IEEE Transactions on Neural Networks & Learning Systems*, vol. 99, pp. 1–15, 2017.

[31] K.-K. Xu, H.-D. Yang, and C.-J. Zhu, "A novel extreme learning machine-based Hammerstein-Wiener model for complex nonlinear industrial processes," *Neurocomputing*, vol. 358, pp. 246–254, 2019.

[32] J. K. Ali, "Neural networks: a new tool for the petroleum industry?," in *Proceedings of the European Petroleum Computer Conference*, Aberdeen, UK, March 1994.

Hindawi
Shock and Vibration
Volume 2024, Article ID 9872819, 1 page
https://doi.org/10.1155/2024/9872819

Retraction

Retracted: Parallel-Machine Scheduling with DeJong's Learning Effect, Delivery Times, Rate-Modifying Activity, and Resource Allocation

Shock and Vibration

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] L. Sun, B. Wu, and L. Ning, "Parallel-Machine Scheduling with DeJong's Learning Effect, Delivery Times, Rate-Modifying Activity, and Resource Allocation," *Shock and Vibration*, vol. 2021, Article ID 6687525, 10 pages, 2021.

Hindawi
Shock and Vibration
Volume 2021, Article ID 6687525, 10 pages
https://doi.org/10.1155/2021/6687525

Research Article

Parallel-Machine Scheduling with DeJong's Learning Effect, Delivery Times, Rate-Modifying Activity, and Resource Allocation

Li Sun,[1] **Bin Wu**,[1] **and Lei Ning**[2]

[1]*School of Business, Shanghai Dianji University, Shanghai 201306, China*
[2]*College of Engineering Science and Technology, Shanghai Ocean University, Shanghai 201306, China*

Correspondence should be addressed to Bin Wu; wub@sdju.edu.cn

Received 19 November 2020; Revised 11 December 2020; Accepted 19 December 2020; Published 18 January 2021

Academic Editor: Tangbin Xia

We investigate parallel-machine scheduling with past-sequence-dependent (p-s-d) delivery times, DeJong's learning effect, rate-modifying activity, and resource allocation. Each machine has a rate-modifying activity. We consider two versions of the problem to minimize the sum of the total completion times, the total absolute deviation of job completion times, and the total resource allocation and the sum of the total waiting times, the total absolute deviation of job waiting times, and the total resource allocation, respectively. The problems under our present model can be solved in polynomial time.

1. Introduction

In practice, a finite amount of resource usually is allocated to a job to control its actual processing, which is the so-called scheduling problem with controllable processing times. Researchers in this case have to make two decisions–job sequence and resource allocation simultaneously–which is different from common scheduling problems. These kinds of scheduling problems have attracted a great deal of attention in the last three decades since Vickson. Vickson [1] initiated this field. The resource allocation function usually has two forms including a linear function and a convex function. Liu and Feng [2] address two-machine flowshop scheduling problems in which the processing time of a job is a function of its position in the sequence and its resource allocation. Zhu et al. [3]investigate scheduling problems with a deteriorating and resource-dependent maintenance activity. They show that all the considered problems are polynomially solvable. Liu et al. [4] consider a parallel-machine scheduling problem to minimize the sum of resource consumption and outsourcing cost. Liu et al. [5] consider single-machine scheduling problems which determine the optimal job schedule, due-window location, and resource allocation simultaneously.

In industrial production, machine unavailability periods are very common which is first studied by Lee and Leon. [6]. Motivated by this phenomenon, scheduling with a rate-modifying activity becomes a popular topic in the last decade. Zhu et al. [7] addresses a single-machine scheduling problem with resource allocation and a rate-modifying activity simultaneously. Ji et al. [8] consider single-machine scheduling with a common due-window and a deteriorating rate-modifying activity. Polynomial-time solution algorithms are provided for the corresponding problems. Yang and Yang [9] investigate parallel-machine scheduling problems with multiple rate-modifying activities. Zhu et al. [3] study single-machine scheduling problems with a deteriorating and resource-dependent maintenance activity. Luo [10] addresses a single-machine scheduling problem with a deteriorating rate-modifying activity to minimize the number of tardy jobs. He proposed an optimal polynomial time algorithm. Yu [11] considers an optimal single-machine scheduling with linear deterioration rate and rate-modifying activities.

In modern industry, the manufacturing environment has a great impact on jobs′ processing times. Such an extra time for eliminating the adverse effects between the main

processing and the delivery of a job is viewed as a past-sequence-dependent (p-s-d) delivery time. Koulamas and Kyparisis [12] first introduced p-s-d delivery time into scheduling problem. Liu et al. [13] considered the problem of single-machine scheduling with p-s-d delivery times, which was introduced in Koulamas and Kyparisis [12]. Liu [14] introduced identical parallel-machine scheduling with p-s-d delivery times and the learning effect. Shen and Wu [15] studied single-machine scheduling with p-s-d delivery times and general learning effects.

The workers can acquire experience and improve the production efficiency continuously, and this phenomenon−first discussed by Wright [16]−is called the learning effect in the literature [17]. Wu et al. [18] study some single-machine scheduling problems with elapsed-time-based and position-based learning and forgetting effects. More recent papers that consider scheduling with learning effect include Rostami et al. [19], Zhang et al. [20], Yin et al. [21], Zhang and Wang [22], Toksari and Arik [23], Jiang et al. [24], Cheng et al. [25], Pei et al. [26], Mustu and Eren [27], and Liu and Feng. [2]. The above scheduling model with the position-based learning effect suffers a drawback that job's actual processing time is close to zero when the job's position is sufficiently large in a schedule. Scheduling problem with DeJong's learning effect is proposed, which overcomes the shortcomings in Wright's learning model. Okoowski and Gawiejnowicz [28] consider a parallel-machine scheduling problem with DeJong's learning effect and makespan objective. Ji et al. [29] consider a learning model in scheduling based on DeJong's learning effect. Ji et al. [30] consider parallel-machine scheduling with deteriorating jobs and DeJong's learning effect. They show minimizing the total completion time is polynomially solvable and minimizing the makespan is NP-hard. Throughout the paper, we will consider parallel-machine scheduling problem with DeJong's learning effect.

Scheduling problems concerning multimachine production environments are encountered in many modern manufacturing processes. To the best of our knowledge, scheduling with p-s-d delivery times, DeJong's learning effect, rate-modifying activity, and resource allocation has not been studied in the literature. In this paper, we study two versions of such problems under linear and convex resource consumption and show the problems are polynomially solvable. The remaining part of this paper is organized as follows. In Section 2, we formulate the problem and present some notation and one lemma. We introduce two versions of the problem to minimize the sum of the total completion times, the total absolute deviation of job completion times, and the total resource allocation and the sum of the total waiting times, the total absolute deviation of job waiting times, and the total resource allocation in Section 3. In Section 4, we conclude the paper.

2. Problem Formulation

There are a set of n independent and non-pre-emptive jobs simultaneously available for processing and m identical parallel machines. Each machine can handle one job at a time. With the assumption that $m < n$ throughout the paper, since the problem is trivial, if $m \geq n$, let $p_{ij}\,(p_{ij}^A)$ be the normal (actual) processing time of job J_{ij} and $p_{i[r]}\,(p_{i[r]}^A)$ be the normal (actual) processing time of job $J_{i[r]}$ if it is scheduled in the rth position on machine M_i in a sequence. In view of the study of DeJong's learning model for scheduling, we adopt it in our paper as follows: $p_{i[r]}^A = p_{i[r]}\,(M + (1-M)r^{a_{i[r]}})$, where $a_{i[r]}$ is a nonpositive learning index and $a_{i[r]} < 0$. It is easy to know that if $M = 0$, the model reduces to the classical learning model.

In this paper, we will consider the situation of repairing or upgrading the machine that one rate-modifying activity is allowed on each machine throughout the scheduling to improve the machines production efficiency which is denoted by RMA. A rate-modifying activity (RMA) can be applied to the machine so as to change (usually to decrease) the normal processing times of the jobs. The time p_{ij} of processing job J_{ij} changes after the RMA to $\lambda_{ij}p_{ij}$. The machine will revert to its initial condition, and the learning effect will start anew after the rate-modifying activity. Suppose n_i is the number of jobs located on machine M_i and k_i is the position of the rate-modifying activity on machine M_i. In this paper, we consider two resource consumption functions.

A linear resource consumption function:

$$p_{i[r]}^A = p_{i[r]}\left(M + (1-M)r^{a_{i[r]}}\right) - b_{i[r]}u_{i[r]}, \tag{1}$$

before rate-modifying activity and

$$p_{i[r]}^A = \lambda_{i[r]}p_{i[r]}\left(M + (1-M)\left(r-k_i\right)^{a_{i[r]}}\right) - b_{i[r]}u_{i[r]}, \tag{2}$$

after rate-modifying activity, where $\lambda_{i[r]}$ is the modifying rate to job $J_{i[r]}$ with $0 < \lambda_{i[r]} \leq 1$, $u_{i[r]}$ is the amount of the resource allocated to job $J_{i[r]}$ with $0 \leq u_{i[r]} \leq \overline{u}_{i[r]} < ((\lambda_{i[r]}p_{i[r]})/(b_{i[r]}))$, and $b_{i[r]}$ is the positive compression rate of job $J_{i[r]}$.

A convex resource consumption function:

$$p_{i[r]}^A = \left(\frac{p_{i[r]}\left(M + (1-M)r^{a_{i[r]}}\right)}{u_{i[r]}}\right)^{v}, \tag{3}$$

before rate-modifying activity and

$$p_{i[r]}^A = \left(\frac{\lambda_{i[r]}p_{i[r]}\left(M + (1-M)\left(r-k_i\right)^{a_{i[r]}}\right)}{u_{i[r]}}\right)^{v}, \tag{4}$$

after rate-modifying activity, where v is a positive constant. The rate-modifying activity duration is a linear function of its starting time which is represented by $f(t) = \beta + \sigma t$, where $\beta > 0$ is the basic rate-modifying activity time, $\sigma > 0$ is a rate-modifying activity factor, and t is the starting time of the rate-modifying activity operation. The starting time of the rate-modifying activity is not known in advance, and it can be scheduled immediately after completing the processing of any job.

As in [12], the processing of job $J_{i[r]}$ must be followed by the p-s-d delivery time $q_{i[r]}$, which can be calculated as

$$\begin{aligned} q_{i[1]} &= 0, \\ q_{i[r]} &= \gamma W_{i[r]} = \gamma \sum_{l=1}^{r-1} p_{i[l]}^A, \end{aligned} \tag{5}$$

before rate-modifying activity and

$$q_{i[r]} = \gamma W_{i[r]} = \gamma\left(\sum_{l=1}^{r-1} p^A_{i[l]} + f(t)\right), \tag{6}$$

after rate-modifying activity, where $\gamma \geq 0$ is a normalizing constant and $W_{i[r]}$ denotes the waiting time of job $J_{i[r]}$. As usual, the postprocessing operation of any job $J_{i[l]}$ modelled by its delivery time $q_{i[l]}$ is performed off-line. Hence, it is not affected by the availability of the machine, and it can be implemented immediately upon completion of the main operation, and we have

$$\begin{aligned} C_{i[1]} &= p_{i[1]}, \\ C_{i[j]} &= W_{i[j]} + p^A_{i[j]} + q_{i[j]} = (1+\gamma)W_{i[j]} + p^A_{i[j]}, \end{aligned} \tag{7}$$

where $C_{i[j]}$ denotes the completion time of job $J_{i[j]}$.

Let denote the p-s-d delivery time by q_{psd}. In addition, we denote TADC_i the total absolute deviation of job completion times and TADW_i the total absolute deviation of job waiting times on machine M_i, i.e., $\text{TADC}_i = \sum_{l=1}^{n_i}\sum_{k=l}^{n_i} |C_{i[l]} - C_{i[k]}|$ and $\text{TADW}_i = \sum_{l=1}^{n_i}\sum_{k=l}^{n_i} |W_{i[l]} - W_{i[k]}|$. Let TC_i indicates the job's total processing times on machine M_i and TW_i indicates the job's total waiting times on machine M_i, i.e., $\text{TC}_i = \sum_{r=l}^{n_i} C_{i[r]}$ and $\text{TW}_i = \sum_{r=l}^{n_i} W_{i[r]}$. We will try to find the optimal job sequence, the optimal RMA, and the optimal resource consumption such that the following cost functions are minimized:

$$\begin{aligned} Z_1 &= \alpha_1 \sum_{i=1}^{m} \text{TC}_i + \delta_1 \sum_{i=1}^{m} \text{TADC}_i + \sum_{i=1}^{m}\sum_{j=1}^{n_i} G_{ij}u_{ij}, \\ Z_2 &= \alpha_2 \sum_{i=1}^{m} \text{TW}_i + \delta_2 \sum_{i=1}^{m} \text{TADW}_i + \sum_{i=1}^{m}\sum_{j=1}^{n_i} G_{ij}u_{ij}, \end{aligned} \tag{8}$$

where $\alpha_1, \alpha_2, \delta_1, \delta_2 > 0$ represent the per unit time contribution for the total processing time, the total waiting time, the total absolute deviation of job completion times, and the total absolute deviation of job waiting times on machine M_i with $\alpha_1 > 0$, $\alpha_2 > 0$, $\delta_1 > 0$, and $\delta_2 > 0$. G_{ij} is the per unit time cost associated with resource allocation. Let DJLR denote DeJong's learning effect and linear resource consumption and DJCR denote DeJong's learning effect and convex resource consumption. Using the three-field notation introduced by Graham et al., for scheduling problems, we denote the two versions of the problems as

$$\begin{aligned} &P_m\left|q_{\text{psd}}, \text{DJLR}, \text{RMA}\right|Z, \\ &P_m\left|q_{\text{psd}}, \text{DJCR}, RMA\right|Z, \\ &Z \in \{Z_1, Z_2\}. \end{aligned} \tag{9}$$

We first present some notation and one lemma before the main results. On machine M_i, if the number of jobs n_i and the position of the job preceding the rate-modifying activity k_i are known in advance, then the job's completion times and the job's waiting times on machine M_i are as follows:

$$\begin{aligned} W_{i[1]} &= 0, \\ C_{i[1]} &= p^A_{i[1]}, \\ &\cdots, \\ W_{i[k_i]} &= p^A_{i[1]} + \cdots + p^A_{i[k_i-1]}, \\ C_{i[k_i]} &= (1+\gamma)\left(p^A_{i[1]} + p^A_{i[2]} + \cdots + p^A_{i[k_i-1]}\right) + p^A_{i[k_i]}, \\ f(t) &= \beta + \sigma\left(p^A_{i[1]} + p^A_{i[2]} + \cdots + p^A_{i[k_i]}\right), \\ W_{i[k_i+1]} &= \beta + (1+\sigma)\left(p^A_{i[1]} + \cdots + p^A_{i[k_i]}\right), \\ C_{i[k_i+1]} &= (1+\gamma)\left(\beta + (1+\sigma)\left(p^A_{i[1]} + \cdots + p^A_{i[k_i]}\right)\right) \\ &\quad + p^A_{i[k_i+1]}, \\ &\cdots, \\ W_{i[n_i]} &= \beta + (1+\sigma)\left(p^A_{i[1]} + \cdots + p^A_{i[k_i]}\right) + p^A_{i[k_i+1]} \cdots \\ &\quad + p^A_{i[n_i-1]}, \\ C_{i[n_i]} &= (1+\gamma)\left(\beta + (1+\sigma)\left(p^A_{i[1]} + \cdots + p^A_{i[k_i]}\right)\right. \\ &\quad \left. + p^A_{i[k_i+1]} + \cdots + p^A_{i[n_i-1]}\right) + p^A_{i[n_i]}. \end{aligned} \tag{10}$$

For the linear case,

$$\begin{aligned} p^A_{i[r]} &= p_{i[r]}\left(M + (1-M)r^{a_{i[r]}}\right) - b_{i[r]}u_{i[r]}, \quad \text{if } r \leq k_i, \\ p^A_{i[r]} &= \lambda_{i[r]}p_{i[r]}\left(M + (1-M)(r-k_i)^{a_{i[r]}}\right) - b_{i[r]}u_{i[r]}, \quad \text{if } r \geq k_i. \end{aligned} \tag{11}$$

For the convex case,

$$\begin{aligned} p^A_{i[r]} &= \left(\frac{p_{i[r]}\left(M + (1-M)r^{a_{i[r]}}\right)}{u_{i[r]}}\right)^{v}, \quad \text{if } r \leq k_i, \\ p^A_{i[r]} &= \left(\frac{\lambda_{i[r]}p_{i[r]}\left(M + (1-M)(r-k_i)^{a_{i[r]}}\right)}{u_{i[r]}}\right)^{v}, \quad \text{if } r \geq k_i. \end{aligned} \tag{12}$$

Let $P(n,m,k) = (n_1, n_2, \ldots, n_m; k_1, k_2, \ldots, k_m)$ denote an allocation vector. We provide a lemma concerning an upper bound on the number of $P(n,m,k)$ vectors.

Lemma 1. *The number of $P(n,m,k)$ vectors is bounded from above by $((n+1)^{2m-1})/m!$.*

Proof. See the work of Ma et al. [31].

3. Cases with Linear Resource Consumption Function

3.1. The Problem $P_m|q_{psd}, DJLR, RMA|Z_1$. In this section, we introduce the problem to minimize the sum of total completion times and total absolute deviation of job completion times with resource consumption on all the machines. For machine M_i, from the above analysis, we calculate the total completion times and the total absolute deviation of job completion times on this machine as follows:

$$\mathrm{TC}_i = (n_i - k_i)(1+\gamma)\beta + \sum_{h=1}^{k_i}\left(1 + (k_i - h)(1+\gamma) + (n_i - k_i)(1+\gamma)(1+\sigma)\right)p_{i[h]}^A + \sum_{h=k_i+1}^{n_i}\left(1 + (n_i - h)(1+\gamma)\right)p_{i[h]}^A,$$

$$\mathrm{TADC}_i = \sum_{h=k_i+1}^{n_i}(2h - 1 - n_i)(1+\gamma)\beta + \sum_{h=1}^{k_i}\left((2h - 1 - n_i) + \sum_{l=h+1}^{k_i}(2l - 1 - n_i)(1+\gamma) + \sum_{l=k_i+1}^{n_i}(2l - 1 - n_i)(1+\gamma)(1+\sigma)\right)p_{i[h]}^A$$
$$+ \sum_{h=k_i+1}^{n_i-1}\left((2h - 1 - n_i) + \sum_{l=h+1}^{n_i}(2l - 1 - n_i)(1+\gamma)\right)p_{i[h]}^A + (n_i - 1)p_{i[n_i]}^A. \tag{13}$$

Hence, the sum of total completion times and total absolute deviation of job completion times with resource consumption on all the machines is

$$\begin{aligned}
&\alpha_1\sum_{i=1}^{m}\mathrm{TC}_i + \delta_1\sum_{i=1}^{m}\mathrm{TADC}_i + \sum_{i=1}^{m}\sum_{j=1}^{n_i}G_{ij}u_{ij}\\
&= (1+\gamma)\beta\left(\sum_{i=1}^{m}\alpha_1(n_i - k_i) + \sum_{i=1}^{m}\sum_{h=k_i+1}^{n_i}\delta_1(2h - 1 - n_i)\right) + \sum_{i=1}^{m}\sum_{h=1}^{k_i}(\alpha_1\\
&\quad\cdot\left(1 + (k_i - h)(1+\gamma) + (n_i - k_i)(1+\gamma)(1+\sigma)\right)\\
&\quad + \delta_1\left((2h - 1 - n_i) + \sum_{l=h+1}^{k_i}(2l - 1 - n_i)(1+\gamma) + \sum_{l=k_i+1}^{n_i}(2l - 1 - n_i)(1+\gamma)(1+\sigma)\right)\Bigg)p_{i[h]}^A\\
&\quad + \sum_{i=1}^{m}\sum_{h=k_i+1}^{n_i}\left(\alpha_1\left(1 + (1+\gamma)(n_i - h)\right) + \delta_1\left((2h - 1 - n_i) + \sum_{l=h+1}^{n_i}(2l - 1 - n_i)(1+\gamma)\right)\right)p_{i[h]}^A\\
&\quad + \sum_{i=1}^{m}(\alpha_1 + (n_i - 1)\delta_1)p_{i[n_i]}^A + \sum_{i=1}^{m}\sum_{h=1}^{n_i}G_{i[h]}u_{i[h]}.
\end{aligned} \tag{14}$$

Let

$$A_1 = (1+\gamma)\beta\left(\sum_{i=1}^{m}\alpha_1(n_i-k_i)+\sum_{i=1}^{m}\sum_{h=k_i+1}^{n_i}\delta_1(2h-1-n_i)\right),$$

$$w_{i[h]} = \begin{cases} \alpha_1\left(1+(k_i-h)(1+\gamma)+(n_i-k_i)(1+\gamma)(1+\sigma)\right)+\delta_1\big(\ (2h-1-n_i) \\ +\sum_{l=h+1}^{k_i}(2l-1-n_i)(1+\gamma)+\sum_{l=k_i+1}^{n_i}(2l-1-n_i)(1+\gamma)(1+\sigma)\big), \\ \qquad i=1,2,\ldots,m, h=1,2,\ldots,k_i, \\ \alpha_1\left(1+(1+\gamma)(n_i-h)\right)+\delta_1\big(\ (2h-1-n_i) \\ +\sum_{l=h+1}^{n_i}(2l-1-n_i)(1+\gamma)\big), \\ \qquad i=1,2,\ldots,m, h=k_i+1,k_i+2,\ldots,n_i-1, \\ \alpha_1+(n_i-1)\delta_1 \\ \qquad i=1,2,\ldots,m, h=n_i. \end{cases} \tag{15}$$

Thus,

$$\begin{aligned} &\alpha_1\sum_{i=1}^{m}\mathrm{TC}_i+\delta_1\sum_{i=1}^{m}\mathrm{TADC}_i+\sum_{i=1}^{m}\sum_{j=1}^{n_i}G_{ij}u_{ij} \\ &= A_1+\sum_{i=1}^{m}\sum_{h=1}^{k_i}w_{i[h]}p_{i[h]}\left(M+(1-M)h^{a_{i[h]}}\right) \\ &\quad+\sum_{i=1}^{m}\sum_{h=k_i+1}^{n_i}w_{i[h]}\lambda_{i[h]}p_{i[h]}\left(M+(1-M)\left(h-k_i\right)^{a_{i[h]}}\right) \\ &\quad+\sum_{i=1}^{m}\sum_{h=1}^{n_i}\left(G_{i[h]}-w_{i[h]}b_{i[h]}\right)u_{i[h]}. \end{aligned} \tag{16}$$

From the above equation, for any job sequence, the optimal resource allocation for a job depends on the sign of $G_{i[h]}-w_{i[h]}b_{i[h]}$. If $G_{i[h]}-w_{i[h]}b_{i[h]}$ is negative, the maximum feasible amount of the resource should be allocated to job $J_{i[h]}$, if $G_{i[h]}-w_{i[h]}b_{i[h]}$ is positive, no resource should be allocated to job $J_{i[h]}$, and if $G_{i[h]}-w_{i[h]}b_{i[h]}$ is equal to zero, any of value of resource consumption will not affect the total cost. Let $u^*_{i[h]}$ denote the optimal resource allocation for job $J_{i[h]}$, where

$$u_{i[h]} = \begin{cases} \overline{u}_{i[h]}, & \text{if } G_{i[h]}-w_{i[h]}b_{i[h]}<0, \\ u_0\in[0,\overline{u}_{i[h]}], & \text{if } G_{i[h]}-w_{i[h]}b_{i[h]}=0, \\ 0, & \text{if } G_{i[h]}-w_{i[h]}b_{i[h]}>0. \end{cases} \tag{17}$$

From (17), we can obtain the optimal resource allocation for any given optimal sequence.

Since A_1 is a constant, when n_i and k_i is given, we can express the problem as the following assignment problem:

$$\begin{aligned} &F_1 = A_1+\min\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{h=1}^{n_i}\theta_{ij[h]}x_{ijh}, \\ &(AP_1)\ s.t.\ \sum_{j=1}^{n}x_{ijh}=1,\quad i=1,2,\ldots,m, h=1,2,\ldots,n_i, \\ &\sum_{i=1}^{m}\sum_{h=1}^{n_i}x_{ijh}=1,\quad j=1,2,\ldots,n, \\ &x_{ijh}=0 \text{ or } 1, i=1,2,\ldots,m, h=1,2,\ldots,n_i, j=1,2,\ldots,n, \end{aligned} \tag{18}$$

where

$$\theta_{ij[h]} = \begin{cases} w_{i[h]}p_{ij[h]}+(G_{ij}-w_{i[h]}b_{ij})\overline{u}_{ij}, & \text{if } G_{ij}-w_{i[h]}b_{ij}<0, \\ w_{i[h]}\overline{p}_{ij[h]}, & \text{if } G_{ij}-w_{i[h]}b_{ij}\geq 0, \end{cases}$$

$$p_{ij[h]} = \begin{cases} p_{ij}(M+(1-M)h^{a_{ij}}), & i=1,2,\ldots,m, j=1,2,\ldots,n, h=1,2,\ldots,k_i, \\ \lambda_{ij}p_{ij}(M+(1-M)(h-k_i)^{a_{ij}}), & i=1,2,\ldots,m, j=1,2,\ldots,n, h=k_i+1,k_i+2,\ldots,n_i. \end{cases} \tag{19}$$

Consequently, when $P(n,m,k)$ vector is given, optimal job scheduling and optimal resource allocation are given by Algorithm 1.

Since the $P(n,m,k)$ vector is given, we know that the problem can be solved in $O(n^3)$ time. Together with Lemma 1, it is easy to obtain the following theorem.

Step 1: jobs are scheduled by (AP_1)
Step 2: optimal job resource allocation is calculated by formula (17)

Algorithm 1: Algorithm to solve the problem of minimizing the sum of total completion times and total absolute deviation of job completion times with linear resource consumption.

Theorem 1. *The problem $P_m|q_{psd}, DJLR, RMA|Z_1$ can be solved in $O(n^{2m+2})$ time.*

3.2. The Problem $P_m|q_{psd}, DJLR, RMA|Z_2$. In this section, we study the problem to minimize the sum of total waiting times and total absolute deviation of job waiting times with resource consumption on all the machines. For machine M_i, we compute the total waiting times and the total absolute deviation of job waiting times on this machine as follows:

$$\mathrm{TW}_i = (n_i - k_i)\beta + \sum_{h=1}^{k_i}\left((n_i - k_i)(1+\sigma) + k_i - h\right)p_{i[h]}^A + \sum_{h=k_i+1}^{n_i}(n_i - h)p_{i[h]}^A,$$

$$\mathrm{TADW}_i = (k_i - 1)(n_i - k_i)\beta + \sum_{h=1}^{k_i}\left((k_i - h)(n_i - k_i)\sigma + (h-1)(k_i - h + (n_i - k_i)(1+\sigma))\right)p_{i[h]}^A + \sum_{h=k_i+1}^{n_i} k_i(n_i - h)p_{i[h]}^A. \tag{20}$$

Hence, the sum of total waiting times and total absolute deviation of job waiting times with resource consumption on all the machines is

$$\begin{aligned}
&\alpha_2\sum_{i=1}^{m}\mathrm{TW}_i + \delta_2\sum_{i=1}^{m}\mathrm{TADW}_i + \sum_{i=1}^{m}\sum_{j=1}^{n_i}G_{ij}u_{ij}\\
&= \sum_{i=1}^{m}(\alpha_2 + (k_i - 1)\delta_2)(n_i - k_i)\beta\\
&\quad + \sum_{i=1}^{m}\sum_{h=1}^{k_i}\left(\alpha_2\left((n_i - k_i)(1+\sigma) + k_i - h\right)\right.\\
&\quad + \delta_2\left((k_i - h)(n_i - k_i)\sigma + (h-1)\right.\\
&\quad \left.\left.\cdot\,(k_i - h + (n_i - k_i)(1+\sigma))\right)\right)p_{i[h]}^A\\
&\quad + \sum_{i=1}^{m}\sum_{h=k_i+1}^{n_i}(\alpha_2 + k_i\delta_2)(n_i - h)p_{i[h]}^A + \sum_{i=1}^{m}\sum_{h=1}^{n_i}G_{i[h]}u_{i[h]}.
\end{aligned} \tag{21}$$

Let

$$A_2 = \sum_{i=1}^{m}(\alpha_2 + (k_i - 1)\delta_2)(n_i - k_i)\beta,$$

$$\varphi_{i[h]} = \begin{cases}
\alpha_2\left((n_i - k_i)(1+\sigma) + k_i - h\right) + \delta_2\left((k_i - h)(n_i - k_i)\sigma\right.\\
\left. + (h-1)(k_i - h + (n_i - k_i)(1+\sigma))\right),\\
\qquad i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i,\\
(\alpha_2 + k_i\delta_2)(n_i - h),\\
\qquad i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i.
\end{cases} \tag{22}$$

Thus,

$$\begin{aligned}
&\alpha_2\sum_{i=1}^{m}\mathrm{TW}_i + \delta_2\sum_{i=1}^{m}\mathrm{TADW}_i + \sum_{i=1}^{m}\sum_{j=1}^{n_i}G_{ij}u_{ij}\\
&= A_2 + \sum_{i=1}^{m}\sum_{h=1}^{k_i}\varphi_{i[h]}p_{i[h]}\left(M + (1-M)h^{a_{i[h]}}\right)\\
&\quad + \sum_{i=1}^{m}\sum_{h=k_i+1}^{n_i}\lambda_{i[h]}\varphi_{i[h]}p_{i[h]}\left(M + (1-M)(h - k_i)^{a_{i[h]}}\right)\\
&\quad + \sum_{i=1}^{m}\sum_{h=1}^{n_i}\left(G_{i[h]} - \varphi_{i[h]}b_{i[h]}\right)u_{i[h]}.
\end{aligned} \tag{23}$$

For any job sequence, the optimal resource allocation for a job depends on the sign of $G_{i[h]} - \varphi_{i[h]}b_{i[h]}$. Let $u_{i[h]}^*$ denote the optimal resource allocation for job $J_{i[h]}$, where

$$u_{i[h]} = \begin{cases}
\overline{u}_{i[h]}, & \text{if } G_{i[h]} - \varphi_{i[h]}b_{i[h]} < 0,\\
u_0 \in [0, \overline{u}_{i[h]}], & \text{if } G_{i[h]} - \varphi_{i[h]}b_{i[h]} = 0,\\
0, & \text{if } G_{i[h]} - \varphi_{i[h]}b_{i[h]} > 0.
\end{cases} \tag{24}$$

From (24), we can get the optimal resource allocation for any given optimal sequence.

Accordingly, when n_i and k_i is given, we can indicate the problem as the following assignment problem:

Step 1: jobs are scheduled by (AP_2)
Step 2: optimal job resource allocation is calculated by formula (24)

Algorithm 2: Algorithm to solve the problem of minimizing the sum of total waiting times and total absolute deviation of job waiting times with linear resource consumption.

$$F_2 = A_2 + \min \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{h=1}^{n_i} \rho_{ij[h]} y_{ijh},$$

$$(AP_2)\ s.t. \sum_{j=1}^{n} y_{ijh} = 1, \quad i = 1,2,\ldots,m, h = 1,2,\ldots,n_i,$$

$$\sum_{i=1}^{m}\sum_{h=1}^{n_i} y_{ijh} = 1, \quad j = 1,2,\ldots,n, y_{ijh} = 0 \text{ or } 1, \ i = 1,2,\ldots,m, h = 1,2,\ldots,n_i, j = 1,2,\ldots,n, \tag{25}$$

where

$$\rho_{ij[h]} = \begin{cases} \varphi_{i[h]} p_{ij[h]} + \left(G_{ij} - \varphi_{i[h]} b_{ij}\right)\overline{u}_{ij}, & \text{if } G_{ij} - \varphi_{i[h]} b_{ij} < 0, \\ \varphi_{i[h]} p_{ij[h]}, & \text{if } G_{ij} - \varphi_{i[h]} b_{ij} \geq 0, \end{cases}$$

$$p_{ij[h]} = \begin{cases} p_{ij}\left(M + (1-M)h^{a_{ij}}\right), & i = 1,2,\ldots,m, \\ & j = 1,2,\ldots,n, h = 1,2,\ldots,k_i, \\ \lambda_{ij} p_{ij}\left(M + (1-M)\left(h - k_i\right)^{a_{ij}}\right), & \\ i = 1,2,\ldots,m, j = 1,2,\ldots,n, h = k_i + 1, k_i + 2, \ldots, n_i. \end{cases} \tag{26}$$

Hence, when $P(n,m,k)$ vector is given, optimal job scheduling and optimal resource allocation are given by Algorithm 2.

Thus, when the $P(n,m,k)$ vector is given, the problem can be solved in $O(n^3)$ time. Together with Lemma 1, we have the following theorem.

Theorem 2. *The problem $P_m|q_{psd}, DJLR, RMA|Z_2$ can be solved in $O(n^{2m+2})$ time.*

4. Cases with Convex Resource Consumption Function

In this section, we will consider the problems under convex resource consumption function, i.e.,

$$P_m\left|q_{\text{psd}}, \text{DJCR}, \text{RMA}\right|Z, \quad Z \in \{Z_1, Z_2\}. \tag{27}$$

Let

$$\overline{p}_{i[h]} = \begin{cases} p_{i[h]}\left(M + (1-M)h\right)^{a_{i[h]}}, & i = 1,2,\ldots,m, h = 1,2,\ldots,k_i, \\ \lambda_{i[h]} p_{i[h]}\left(M + (1-M)\left(h - k_i\right)^{a_{i[h]}}\right), & i = 1,2,\ldots,m, h = k_i + 1, k_i + 2, \ldots, n_i. \end{cases} \tag{28}$$

Similar to the analysis of problem $P_m|q_{\text{psd}}, \text{DJLR}, \text{RMA}|Z_1$, if n_i and k_i is given, we calculate the problem to minimize the sum of total completion times and total absolute deviation of job completion times with convex resource consumption as follows:

$$\begin{aligned} H_1 &= \alpha_1 \sum_{i=1}^{m} \text{TC}_i + \delta_1 \sum_{i=1}^{m} \text{TADC}_i + \sum_{i=1}^{m}\sum_{j=1}^{n_i} G_{ij} u_{ij} \\ &= A_1 + \sum_{i=1}^{m}\sum_{h=1}^{n_i} w_{i[h]}\left(\frac{\overline{p}_{i[h]}}{u_{i[h]}}\right)^{v} + \sum_{i=1}^{m}\sum_{h=1}^{n_i} G_{i[h]} u_{i[h]}, \end{aligned} \tag{29}$$

where

$$A_1 = (1+\gamma)\beta\left(\sum_{i=1}^{m} \alpha_1 (n_i - k_i) + \sum_{i=1}^{m}\sum_{h=k_i+1}^{n_i} \delta_1 (2h - 1 - n_i)\right),$$

$$w_{i[h]} = \begin{cases} \alpha_1\left(1 + (k_i - h)(1+\gamma) + (n_i - k_i)(1+\gamma)(1+\sigma)\right) + \delta_1\left((2h - 1 - n_i)\right. & \\ \left. + \sum_{l=h+1}^{k_i}(2l - 1 - n_i)(1+\gamma) + \sum_{l=k_i+1}^{n_i}(2l - 1 - n_i)(1+\gamma)(1+\sigma)\right), & i = 1,2,\ldots,m, h = 1,2,\ldots,k_i, \\ \alpha_1\left(1 + (1+\gamma)(n_i - h)\right) + \delta_1\left((2h - 1 - n_i) + \sum_{l=h+1}^{n_i}(2l - 1 - n_i)(1+\gamma)\right), & i = 1,2,\ldots,m, h = k_i + 1, k_i + 2, \ldots, n_i - 1, \\ \alpha_1 + (n_i - 1)\delta_1, & i = 1,2,\ldots,m, h = n_i. \end{cases} \tag{30}$$

Step 1: jobs are scheduled by (AP_3)
Step 2: optimal job resource allocation is calculated by formula (31)

Algorithm 3: Algorithm to solve the problem of minimizing the sum of total completion times and total absolute deviation of job completion times with convex resource consumption.

By taking the first derivative of H_1 with respect to $u_{i[h]}$, $i = 1, 2, \ldots, m$ and $h = 1, 2, \ldots, n_i$, equating the result to zero, and solving it for $u_{i[h]}$, we can obtain the optimal resource allocation (denoted by $u^*_{i[h]}$):

$$\frac{\partial H_1}{\partial u_{i[h]}} = -v w_{i[h]} \left(\overline{p}_{i[h]}\right)^v u_{i[h]}^{-v-1} + G_{i[h]} = 0,$$
$$u^*_{i[h]} = \left(\frac{v w_{i[h]}}{G_{i[h]}}\right)^{1/(v+1)} \left(\overline{p}_{i[h]}\right)^{v/(v+1)}. \tag{31}$$

By substituting $u^*_{i[h]}$ into the objective function H_1, we obtain a new unified expression as follows:

$$H_1 = A_1 + \sum_{i=1}^{m} \sum_{h=1}^{n_i} \left(v^{-v/(v+1)} + v^{1/(v+1)}\right) w_{i[h]}^{1/(v+1)} \left(G_{i[h]} \overline{p}_{i[h]}\right)^{v/(v+1)}. \tag{32}$$

Therefore, we can formulate the minimum problem as the following assignment problem:

$$H_1 = A_1 + \min \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{h=1}^{n_i} \xi_{ij[h]} \overline{x}_{ijh},$$
$$(AP_3)\, s.t. \sum_{j=1}^{n} \overline{x}_{ijh} = 1, \quad i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i,$$
$$\sum_{i=1}^{m} \sum_{h=1}^{n_i} \overline{x}_{ijh} = 1, \quad j = 1, 2, \ldots, n, \overline{x}_{ijh} = 0 \text{ or } 1,$$
$$i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i, j = 1, 2, \ldots, n, \tag{33}$$

where

$$\xi_{ij[h]} = \begin{cases} \left(v^{-v/(v+1)} + v^{1/(v+1)}\right) w_{i[h]}^{1/(v+1)} \left(G_{ij} p_{ij} \left(M + (1-M) h^{a_{ij}}\right)\right)^{v/(v+1)}, & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = 1, 2, \ldots, k_i, \\ \left(v^{-v/(v+1)} + v^{1/(v+1)}\right) w_{i[h]}^{1/(v+1)} \left(G_{ij} \lambda_{ij} p_{ij} \left(M + (1-M)(h - k_i)^{a_{ij}}\right)\right)^{v/(v+1)}, & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = k_i + 1, k_i + 2, \ldots, n_i. \end{cases} \tag{34}$$

Consequently, when $P(n, m, k)$ vector is given, optimal job scheduling and optimal resource allocation are given by Algorithm 3.

Together with Lemma 1, we have the following theorem.

Theorem 3. *The problem $P_m|q_{psd}, DJCR, RMA|Z_1$ can be solved in $O(n^{2m+2})$ time.*

Similar to the analysis of problem $P_m|q_{\text{psd}}, \text{DJLR}, \text{RMA}|\alpha_2 \sum_{i=1}^{m} \text{TW}_i + \delta_2 \sum_{i=1}^{m} \text{TADW}_i + \sum_{i=1}^{m} \sum_{j=1}^{n_i} G_{ij} u_{ij}$, if n_i and k_i is given, we calculate the problem to minimize the sum of total waiting times and total absolute deviation of job waiting times with convex resource consumption as follows:

$$H_2 = \alpha_2 \sum_{i=1}^{m} \text{TW}_i + \delta_2 \sum_{i=1}^{m} \text{TADW}_i + \sum_{i=1}^{m} \sum_{j=1}^{n_i} G_{ij} u_{ij}$$
$$= A_2 + \sum_{i=1}^{m} \sum_{h=1}^{n_i} \varphi_{i[h]} \left(\frac{\overline{p}_{i[h]}}{u_{i[h]}}\right)^v + \sum_{i=1}^{m} \sum_{h=1}^{n_i} G_{i[h]} u_{i[h]}, \tag{35}$$

where

$$A_2 = \sum_{i=1}^{m} (\alpha_2 + (k_i - 1)\delta_2)(n_i - k_i)\beta,$$
$$\varphi_{i[h]} = \begin{cases} \alpha_2 \left((n_i - k_i)(1 + \sigma) + k_i - h\right) + \delta_2 ((k_i - h)(n_i - k_i)\sigma \\ +(h - 1)(k_i - h + (n_i - k_i)(1 + \sigma))), & i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i, \\ (\alpha_2 + k_i \delta_2)(n_i - h), & i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i, \end{cases} \tag{36}$$
$$\overline{p}_{i[h]} = \begin{cases} p_{i[h]} (M + (1 - M)h)^{a_{i[h]}}, & i = 1, 2, \ldots, m, h = 1, 2, \ldots, k_i, \\ \lambda_{i[h]} p_{i[h]} (M + (1 - M)(h - k_i)^{a_{i[h]}}), & i = 1, 2, \ldots, m, h = k_i + 1, k_i + 2, \ldots, n_i. \end{cases}$$

Step 1: jobs are scheduled by (AP_4)
Step 2: optimal job resource allocation is calculated by formula (37)

Algorithm 4: Algorithm to solve the problem of minimizing the sum of total waiting times and total absolute deviation of job waiting times with convex resource consumption.

Hence, taking the first derivative of H_2 with respect to $u_{i[h]}$, $i = 1, 2, \ldots, m$ and $h = 1, 2, \ldots, n_i$, equating the result to zero, and solving it for $u_{i[h]}$, we can obtain the optimal resource allocation (denoted by $u^*_{i[h]}$):

$$\frac{\partial H_2}{\partial u_{i[h]}} = -v\varphi_{i[h]}\left(\overline{p}_{i[h]}\right)^v u_{i[h]}^{-v-1} + G_{i[h]} = 0,$$
$$u^*_{i[h]} = \left(\frac{v\varphi_{i[h]}}{G_{i[h]}}\right)^{1/(v+1)} \left(\overline{p}_{i[h]}\right)^{v/(v+1)}. \tag{37}$$

By substituting $u^*_{i[h]}$ into the objective function H_2, we obtain a new unified expression as follows:

$$H_2 = A_2 + \sum_{i=1}^{m} \sum_{h=1}^{n_i} \left(v^{-v/(v+1)} + v^{1/(v+1)}\right) \varphi_{i[h]}^{1/(v+1)} \left(G_{i[h]}\overline{p}_{i[h]}\right)^{v/(v+1)}. \tag{38}$$

Therefore, we can formulate the minimum problem as the following assignment problem:

$$H_2 = A_2 + \min \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{h=1}^{n_i} \eta_{ij[h]} \overline{y}_{ijh},$$
$$(AP_4)\ s.t. \sum_{j=1}^{n} \overline{y}_{ijh} = 1, \quad i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i,$$
$$\sum_{i=1}^{m} \sum_{h=1}^{n_i} \overline{y}_{ijh} = 1, \quad j = 1, 2, \ldots, n, \overline{y}_{ijh} = 0 \text{ or } 1,$$
$$i = 1, 2, \ldots, m, h = 1, 2, \ldots, n_i, j = 1, 2, \ldots, n, \tag{39}$$

where

$$\eta_{ij[h]} = \begin{cases} \left(v^{-v/(v+1)} + v^{1/(v+1)}\right) \varphi_{i[h]}^{1/(v+1)} \left(G_{ij} p_{ij} \left(M + (1-M) h^{a_{ij}}\right)\right)^{v/(v+1)}, & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = 1, 2, \ldots, k_i, \\ \left(v^{-v/(v+1)} + v^{1/(v+1)}\right) \varphi_{i[h]}^{1/(v+1)} \left(G_{ij}\lambda_{ij} p_{ij} \left(M + (1-M)\left(h - k_i\right)^{a_{ij}}\right)\right)^{v/(v+1)}, & i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, h = k_i + 1, k_i + 2, \ldots, n_i. \end{cases} \tag{40}$$

Therefore, when $P(n, m, k)$ vector is given, optimal job scheduling and optimal resource allocation are given by Algorithm 4.

From the above analysis and Lemma 1, we have the following theorem.

Theorem 4. *The problem $P_m|q_{psd}, DJCR, RMA|Z_2$ can be solved in $O(n^{2m+2})$ time.*

5. Conclusions

In this paper, two versions of parallel-machine scheduling problems to minimize the sum of the total completion times, the total absolute deviation of job completion times, and the total resource allocation and the sum of the total waiting times, the total absolute deviation of job waiting times, and the total resource allocation are considered, respectively. We present the problems in this research can be solved polynomially. Future research will be worth extending to multiple rate-modifying activity or other objective scheduling problems.

Data Availability

All data generated or analysed during this study are included in this article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. G. Vickson, "Two single machine sequencing problems involving controllable job processing times," *A I I E Transactions*, vol. 12, no. 3, pp. 258–262, 1980.

[2] Y. Liu and Z. Feng, "Two-machine no-wait flowshop scheduling with learning effect and convex resource-dependent processing times," *Computers & Industrial Engineering*, vol. 75, pp. 170–175, 2014.

[3] H. Zhu, M. Li, and Z. Zhou, "Machine scheduling with deteriorating and resource-dependent maintenance activity," *Computers & Industrial Engineering*, vol. 88, pp. 479–486, 2015.

[4] Z. Liu, W.-C. Lee, and J.-Y. Wang, "Resource consumption minimization with a constraint of maximum tardiness on

Hindawi
Shock and Vibration
Volume 2024, Article ID 9756979, 1 page
https://doi.org/10.1155/2024/9756979

Retraction

Retracted: Complex Behavior of Droplet Transfer and Spreading in Cold Metal Transfer

Shock and Vibration

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] S. Yang, Y. Xing, F. Yang, and J. Cao, "Complex Behavior of Droplet Transfer and Spreading in Cold Metal Transfer," *Shock and Vibration*, vol. 2020, Article ID 6650155, 11 pages, 2020.

Hindawi
Shock and Vibration
Volume 2020, Article ID 6650155, 11 pages
https://doi.org/10.1155/2020/6650155

Research Article

Complex Behavior of Droplet Transfer and Spreading in Cold Metal Transfer

Shuai Yang, Yanfeng Xing, Fuyong Yang, and Juyong Cao

School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

Correspondence should be addressed to Yanfeng Xing; smsmsues@163.com

Received 22 October 2020; Accepted 29 October 2020; Published 18 November 2020

Academic Editor: Tangbin Xia

In intelligent manufacturing, an intelligent control method of welding process is an important process of intelligent welding manufacturing technology (IWMT). Metal transfer is a key factor to control the welding process. Metal transfer and droplet spreading are of vital importance for welding formation. A new theoretical model of cold metal transfer (CMT) in short-circuit transfer mode is proposed in this paper. In this model, the CMT welding process is regarded as a continuous process of arc heating, mass transfer, short-circuit, and spreading, and the relations between these processes are analyzed. The calculation equations used by the model can analyze the welding formation clearly and simplify the complex welding process into continuous physical behavior. The predicted welding width shows good agreement with the measurement results. The mechanism of increased welding width is also comprehensively analyzed. Results have a certain guiding effect on aluminum alloy welding process control.

1. Introduction

In intelligent manufacturing, a weld controller is a typical system of the intelligent welding manufacturing system [1]. Intelligent control of welding dynamic process is one of main problems of intelligent welding technology. One of problems in intelligent control of welding dynamic process is welding pool dynamic process [2]. The short-circuit transfer in the metal transition is controlled through traditional MIG (metal inert gas)/MAG (metal active gas) welding to control the heat input and reduce the welding defects. Cold metal transfer (CMT) is an improvement of short-circuit transfer in GMAW (gas-metal arc welding) process, which has the characteristics of low splash and low-heat input. CMT allows a better microstructure to be obtained than MIG. Feng et al. [3] carried out experiments through CMT technology to weld a 1 mm thick pure aluminum alloy with minimal deflection. Lei et al. [4] welded a 1 mm thick AA6061-T6 by using three welding arc modes of standard, pulsed, and CMT. They found that CMT arc mode produced the fewest welding defects. Cao et al. [5, 6] joined 1.0 mm thick AA6061-T6 to 1.0 mm thick galvanized mild steel using the cold metal transfer method, and the welded joint strength can meet the requirements. These studies confirm that CMT is superior to MIG/MAG in thin-plate welding and dissimilar metal welding. CMT can effectively control the heat affected zone (HAZ) and enhance the weldability of thin metal plates and dissimilar metals.

To ensure the formation of welding, scholars have focused on optimization of CMT welding parameters. Wang et al. [7] studied the influences of parameters such as boost current, arcing current, breaking current, boosting voltage, and wire moving speed in CMT welding on the CMT boosting phase, arcing phase, and short-circuit phase. Chen et al. [8] analyzed the effects of CMT current waveform on welding formation. Kumar [9] investigated the effect of welding current and welding speed on welding formation, dilution, and contact angle during cold metal transfer (CMT) process These studies investigated the control of the welding formation through experiments, but the reasons for the changes in the welding formation were not explained theoretically.

Metal droplet behavior has been a largely underexplored domain. Few theoretical models can describe the relationship between metal droplet behavior and welding formation in the CMT process. Most studies on welding formation

model are based on traditional welding methods, such as GMAW. Murphy et al. [10] modeled the arc plasma and weld pool of MIG welding, and predicted the weld depth and shape. Komen et al. [11] simulated the weld pool and the bead formation of GMAW. The simulation realized the droplet transfer into the molten pool, but the model is only used for the prediction of free-flight transfer. The metal droplet detachment from the wire electrode is affected by various forces. Wu et al. [12] developed a metal transfer model of GMAW, which considered droplet oscillation and detachment. Doodman Tipi et al. [13] studied the detachment of GMAW welding and proposed a method for control of the detachment frequency in automatic GMAW process. Chapuis et al. [14] investigated the metal deposition of GMAW and found the law of the metal spreading. Che et al. [15] simulated the evolution of metal spreading. The model assumes that droplets drip freely from a certain height. These studies use simulation to analyze the effect of droplet transfer on the formation of welding, which is a lot of work, and it is difficult to analyze the relationship between physical behaviors.

Wire melting, droplet oscillation, droplet transfer, molten pool cooling, and droplet solidification are important factors that influence the welding process. Droplet transfer is a significant part of welding formation. Some researchers have studied the metal transfer behavior of welding. Bless [16] performed a physical analysis of the short-circuit transition behavior during the welding process. Ersoy et al. [17] studied the drop oscillation and detachment of droplets on GMA by using a mass spring damping system. Kang et al. [18] proposed a dynamic model that describes the growth and separation of droplets for GMAW, considering the effect of electrode melting to form a tapered electrode. These methods enable us to consider all physical phenomena.

Sufficient attention to current and weld metal surface tension can lead to reduced spatter and improved process control. Scotti et al. [19] studied metal transfer in solid wire GMAW welding and classified metal transfer modes. Huang et al. [20] studied the influence of bypass current on metal transfer. Planckaert et al. [21] proposed a hybrid model to investigate MIG/MAG welding process in short-circuit transfer. The three classes of metal transfer modes are nature metal transfer, controlled metal transfer, and interchangeable metal transfer. These studies have investigated the effect of droplet transfer on weld formation during the GMAW process but did not consider the effect of wire feeding on droplet transition, so they cannot explain welding formation of CMT.

In this study, a new CMT droplet model considering the wire electrode movement of short-circuit mode is proposed. This model combines two continuously physical phenomena, namely, metal transfer and metal spreading. In this model, the relations between each stage are analyzed, including droplet growth and short-circuit. The model predicts the formation of droplets at the wire electrode and their transfer into the molten pool in CMT process. The model is validated using experimental measured metal transfer as well as metal spreading. This model allows us to better understand the relationship between welding parameters and weld formation. Results provide a theoretical basis for the control of metal transfer, and solve the main problem of intelligent welding technology. The work helps to control the welding dynamic process in intelligent welding manufacturing.

2. Experimental Procedure

To study the behavior of the droplet and verify the consistency of the theory, we carried out a series of experiments.

From the aspect of droplet transfer, the theory analyzes the change in the wire feed speed of a single cycle CMT that affects the droplet transfer and the weld width. Based on the model of the welding geometry, the CMT welding experiments were carried out, and results were recorded by a high-speed camera. The diagram of experimental equipment is shown in Figure 1. The experimental data were compared with the model to analyze the effect of droplet transfer on the increase in the melt width.

The experimental setup is shown in Figure 2. The experimental materials were AA6061-T6 aluminum. The size of the test pieces was 150 mm 150 mm 2 mm, and the butt joints were used in experiments.

In the experiment, the high-speed camera was Optronis' CP80-3-M, and the recorded videos were used to accurately observe the dynamic droplet transfer behavior during welding. The frame capture rate of the high-speed camera was 4000 frames per second. The images in the high-speed camera were image-extracted and collected, and a total of 10000 images were obtained. The camera also recorded the welding of CMT in different parameters.

The welding equipment used in the study includes Fronius' TPS4000CMT. The welding wire was made of 1.2 mm ER4043 as the filling material, and the shielding gas was argon with a purity of 99.995%. The welding schematic is shown in Figure 1. During the welding process, the welding torch was always perpendicular to the aluminum plate. To reduce the influence of other factors, the test piece was precleaned before welding. More details of the welding parameters are listed in Table 1.

The welding speed of all experiments was set as 0.48 m/min, and the arc correction was −10%. It is to ensure that the droplet deposition is not affected by welding speed and arc correction. In this study, when the wire feed speed is 5.0 m/min, the welding current used in the experiment obtained through the monitoring board is 118 A.

3. Analytical Modeling of Metal Transfer in CMT

3.1. Controlled Short-Circuit Transfer. Arc welding can be seen as a cycle of arc stage and metal transfer stage. In arc stage, with the ignition of the arc, the temperature rises rapidly, which causes the wire melting to form a liquid droplet on the electrode. As the wire gradually melts, metal transfer occurs and thus the mass flows into droplet. When the droplet does not attach on the electrode, it begins to

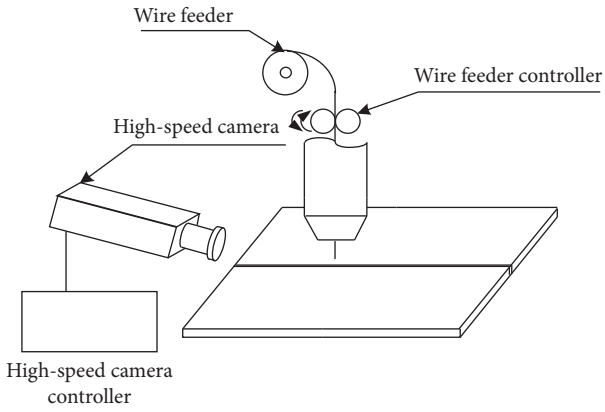


Figure 1: Welding schematic.

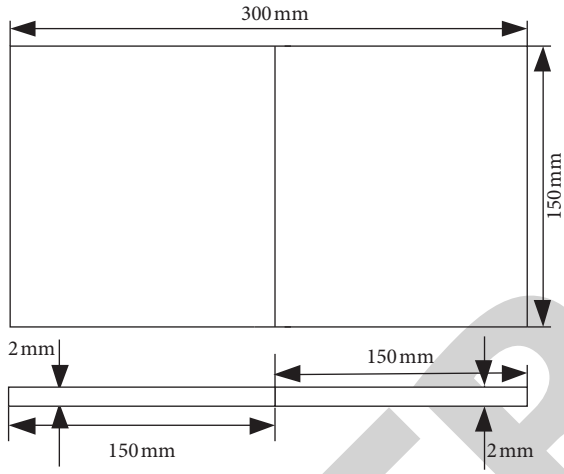


Figure 2: The size of workpiece.

separate from the electrode, then drip into the molten pool, and finally solidify to form a weld to join the metals together.

In the GAMW process, different metal transfer modes can be controlled by current and are affected by wire materials, shielding gas, and process parameters. In the natural state, metal transfer modes can be classified into contact transfer and free-flight transfer. In the free-flight transfer, the droplet is affected by external force and naturally separated from the wire electrode. In this case, arc welding can be divided into the following stages according to the metal transfer behavior: droplet growth stage, droplet separation stage, and droplet spreading stage. In short-circuit transfer, the metal droplet is expanded to the molten pool before separation, and the metal droplet comes in contact with the liquid molten pool. The metal behavior is mainly in the following four stages: droplet growth stage, short-circuit stage, droplet separation stage, and droplet spreading stage.

For some special welding requirements, droplet transfer can be controlled. For example, thin-plate welding and dissimilar metal welding need low-heat input. During the short-circuit transfer mode, droplet slash and heat input become easier to control due to physical contact between the metal droplet and the molten pool. Therefore, some controlled short-circuit transfer technologies have been developed based on short-circuit transfer technology, as reported by Norrish et al. [22].

CMT is related to controlled metal transfer. CMT is based on short-circuit transfer and controls the movement of the wire electrode. The CMT droplet transfer is shown in Figure 3. The transfer of droplet occurs during the short-circuit stage of the welding process cycle, and the movement of the wire helps the droplet transfer into the molten pool. In the necking stage, the movement of the wire is used to assist the separation of the droplet from the wire electrode. In CMT, the controlled short-circuit transfer can reduce the current heat input and the droplet shock, as reported by Pang et al. [23].

3.2. The Growth of Droplet. Welding parameters on CMT mainly affect welding wire melting and droplet transfer. The adjustment of parameters will affect the growth and separation of droplets. In the CMT welding process, the behavior of the metal transfer is influenced by current. In the droplet growth phase, the arc ignites the wire electrode, which is affected by arc heat and Joule heat. The welding wire melts to form droplets at the electrode.

Joule heat can be calculated by

$$Q_J = \int_V \rho(T) I^2 \mathrm{d}V = \overline{\rho} A L I^2, \tag{1}$$

where $\overline{\rho}$ is the average electrical resistivity, L is the electrode extension, and A is the cross sectional area of the electrode. Arc heat can be obtained by

$$Q_A = \left(\frac{3}{2}\frac{kT_a}{e} + V_a + \Phi\right) I, \tag{2}$$

where k is the Boltzmann constant, e is the elementary charge, V_a is the voltage value that characterizes the wire heating by the arc, T_a is the average arc temperature, and Φ is the work function of the electrode material.

Assuming that all heat input is used to melt the wire, the melting rate is the total heat input by the system divided by the heat required to melt the unit mass of the materials, as reported by Planckaert et al. [21]:

$$\frac{\mathrm{d}m_R}{\mathrm{d}t} = \frac{\overline{\rho} A L I^2 + \left((3/2)(kT/e) + V_a + \Phi\right) I}{\int_{T_1}^{T_2} C_p \mathrm{d}T + \Delta H_T + \Delta H_M}, \tag{3}$$

where ΔH_T is the heat of crystalline transition and ΔH_M is the heat of fusion. T_W is the wire temperature. The melting rate can be further simplified as a function of welding current, as reported by Modenesi et al. [24]:

$$\frac{\mathrm{d}m_R}{dt} = \dot{m}_R = M_R = \alpha I + \beta l_s I^2,$$
$$m_R = \frac{4}{3}\pi R^3 \rho, \tag{4}$$

where M_R is the wire melting rate, α is the arc heat coefficient, β is the Joule heat coefficient, m_R is the droplet mass, R is the droplet radius, and ρ is the mass density of metal

Table 1: Experimental parameters used in CMT welding.

Test code	Wire feed speed (m/min)	Welding speed (m/min)	Shielding gas (vol.% Ar)
1	3.0	0.48	99.995
2	3.5	0.48	99.995
3	4.0	0.48	99.995
4	4.5	0.48	99.995
5	5.0	0.48	99.995

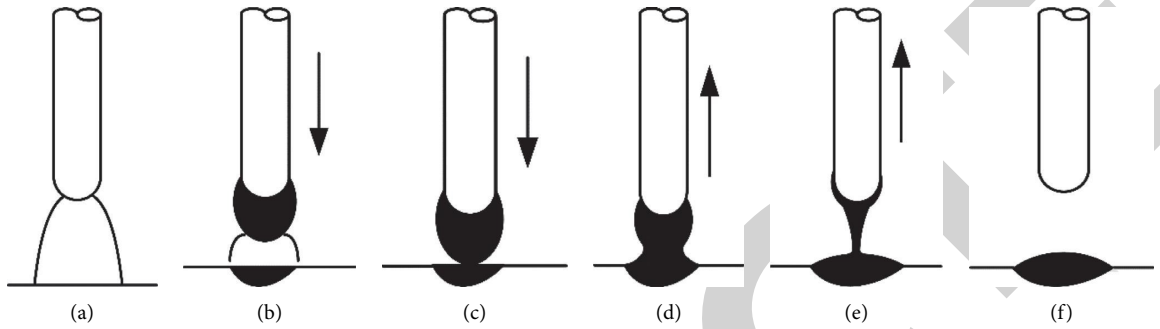


Figure 3: Short-circuit transfer in CMT.

droplet. These values of physical parameters are listed in Table 2.

The wire feed speed is not completely equal to the melting caused by the welding current. The wire feed system controls the wire feed speed. When the wire feed speed increases, it feeds downward, and when it decreases, it draws back:

$$\frac{dl_s}{dt} = v_{wfd} - \frac{dm}{\pi\rho r^2 dt}, \quad (5)$$

where r is the electrode radius, and v_{wfd} is the wire feed speed. When the electrode wire feeding is higher than the wire melting rate, the droplet grows and the wire electrode feed motion helps droplets contact the weld pool.

The growth and separation of droplets can be described by establishing a dynamic model of CMT. The state of the droplets during the transition can be studied using a dynamic model to investigate analyze the influence of the movement of droplet transfer on the melting width of the molten pool.

Assuming that the droplets formed by the welding are spherical droplets, the molten material from the electrodes enters the droplets at a velocity. During the growth phase of the droplet, according to the conservation of momentum, the mass of the droplet changes from m_1 to $m_1 + dm$, and the speed of the droplet relative to the system changes from v_1 to $v_1 + dv$, as shown in Figure 4, where u_1 is the speed of the fluid mass change relative to the droplet, that is, the speed at which the molten wire flows into the droplet.

The growth stage of the droplet can be regarded as a variable mass system. The system meets the momentum theorem. According to the momentum theorem, the liquid metal melted flows into the droplet, and the mass of the droplet increases. The droplet is affected to an external force, and the system momentum changes:

Table 2: Physical parameter values used for calculations.

	Properties (units)	Values
ρ	Density (kg·m^{-3})	2700
A	Area of the electrode (mm^2)	1.44
V_a	Arc voltage (V)	18
α	Arc heating parameter (kg/As)	7.992×10^{-11}
β	Joule heating parameter (kg/mA^{2}s)	1.252×10^{-11}
γ	Surface tension (N/m)	1.2

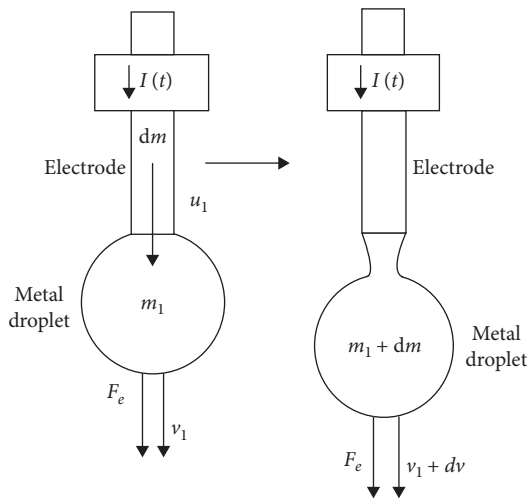


Figure 4: Variable mass system of droplet growth.

$$dp = p_2 - p_1. \quad (6)$$

The impulse of the combined external force on an object is equal to the increase in its momentum, and the equation can be written as

$$\begin{aligned} F_e\mathrm{d}t &= (m_1 + \mathrm{d}m)(v_1 + \mathrm{d}v) - (m_1 v_1 + u_1 \mathrm{d}m), \\ F_e\mathrm{d}t &= m_1 v_1 + v_1 \mathrm{d}m + m_1 \mathrm{d}v + \mathrm{d}m\mathrm{d}v \\ &\quad - v_1 \mathrm{d}m - u_1 \mathrm{d}m - m_1 v_1, \\ F_e\mathrm{d}t &= m_1 \mathrm{d}v + \mathrm{d}m\mathrm{d}v - u_1 \mathrm{d}m, \\ F_e\mathrm{d}t + u_1 \mathrm{d}m &= (m_1 + \mathrm{d}m)\mathrm{d}v. \end{aligned} \tag{7}$$

The conservation of momentum for the metal droplet can be written as

$$m(t)\frac{\mathrm{d}v}{\mathrm{d}t} = F_e + u_1\frac{\mathrm{d}m}{\mathrm{d}t}, \tag{8}$$

where $m(t) = m_R = \int M_R \mathrm{d}t$, F_e is the external forces, and $u_1 = (M_R/4\pi\rho r^2)$.

The external forces that affect droplets are Lorentz force, surface tension, and gravitational force, as reported by Ersoy et al. [17]. In the CMT process, the electrode feed force also needs to be considered, as shown in Figure 5. In the short-circuit stage, the wire feeding system controls the wire drawing to assist the liquid bridge break and metal transfer to the molten pool:

$$\begin{aligned} F_{\mathrm{em}} &= \frac{\mu_0}{4\pi} I^2 P, \\ F_s &= 2\pi R\gamma, \\ F_g &= \frac{4}{3}\pi R^3 \rho g, \\ P &= \left[\ln\frac{R\sin\theta_z}{r} - \frac{1}{4} - \frac{1}{1-\cos\theta_z} + \frac{2}{(1-\cos\theta_z)^2}\ln\frac{2}{1+\cos\theta_z}\right], \end{aligned} \tag{9}$$

where γ is the surface tension coefficient, F_{em} is the Lorentz force, F_s is the surface tension, F_g is the gravitational force, F_f is the electrode feed force, μ_0 is the permeability of free space, R is the droplet radius, r is the electrode radius, and θ_z is the conduction zone radius.

In this model, the droplet displacement caused by the increase in the droplet mass is small, so the liquid bridge shrinkage is ignored in the growth stage. In this case, the liquid bridge radius is equal to the wire radius. In the droplet growth stage, the wire feed force is 0; when the current level is high, external forces will change the droplet transfer mode, which becomes a combination of short-circuit transitions and other transitional forms.

When droplet transfer occurs, the droplet transfer is mainly realized by electrode feeding, as shown in Figure 6. In thin-plate welding, less current is required, the external forces are smaller, the droplets are always attached to the wire electrode, and the transition of the droplet to the molten pool requires electrode movement.

Short-circuit transfer is affected by wire feed and droplet size. As shown in Figure 7, d is the initial distance between the welding wire and the molten pool, the wire electrode length is l_w, Δl_w is the amount of change in wire length, the arc length is l_a, the distance between the droplet and the wire electrode is d_w, and the displacement of the droplet is $D(t)$:

$$D(t) = R(t) + d_w + \Delta l_w. \tag{10}$$

When the value of $D(t)$ increases to be equal to the value of d, a short-circuit transfer occurs, and the condition of short-circuit is

$$D(t) = d. \tag{11}$$

In the short-circuit stage, the wire feeds and contacts the molten pool, and the arc length is 0. The radius of the droplet when a short-circuit transition occurs is recorded as R_i, which is used for subsequent calculations.

Equations (8) and (10) can be combined to obtain the radius of droplet and the displacement of electrode. The calculation results are shown in Figure 8, from which the motion of the droplet can be analyzed.

3.3. Separation of Electrode and Metal Droplet. When the wire is pumped back, the CMT process enters into the necking stage. As shown in Figure 9, when the droplet starts to contact the surface of the base material, the meniscus liquid surface is formed. In this stage, the electrode and the molten pool are connected by a metal liquid bridge.

As the wire is pumped back, the liquid bridge begins to shrink. According to the research of Planckaert, the pressure of the liquid bridge is calculated by

$$P_B = \frac{\mu_0 I^2}{8\pi^2 R_1^2} + \gamma\left(\frac{1}{R_1} + \frac{1}{R_2}\right). \tag{12}$$

When the value of P_B is greater than 0, the liquid bridge is broken, where γ is the surface tension of the liquid bridge. According to the momentum theorem of the variable mass system, the residual mass remaining on the wire electrode can be obtained, which will not be further discussed in this paper.

The flow velocity of the metal bridge flowing into the molten pool surface is calculated by

$$v = \sqrt{\frac{2}{\rho}(P_B + \rho g h)}, \tag{13}$$

where h is the distance from the center of liquid bridge to the surface of molten pool, which is affected by the wire extension.

3.4. The Spreading of Metal Droplet on the Surface of the Workpiece. The droplet that transitions into the molten pool will deform quickly, and in this stage, the motion of the droplet can be simplified as droplet spreading, as reported by Sivakumar et al. [25].

The droplet has an initial velocity that strikes the workpiece. In the CMT process, the droplet is always attached to the wire electrode when impacting the workpiece. Due to the influence of short-circuit transfer, the droplet is not affected by the acceleration of gravity before contacting the workpiece surface. The initial velocity of the droplet is equal to the wire feed speed.

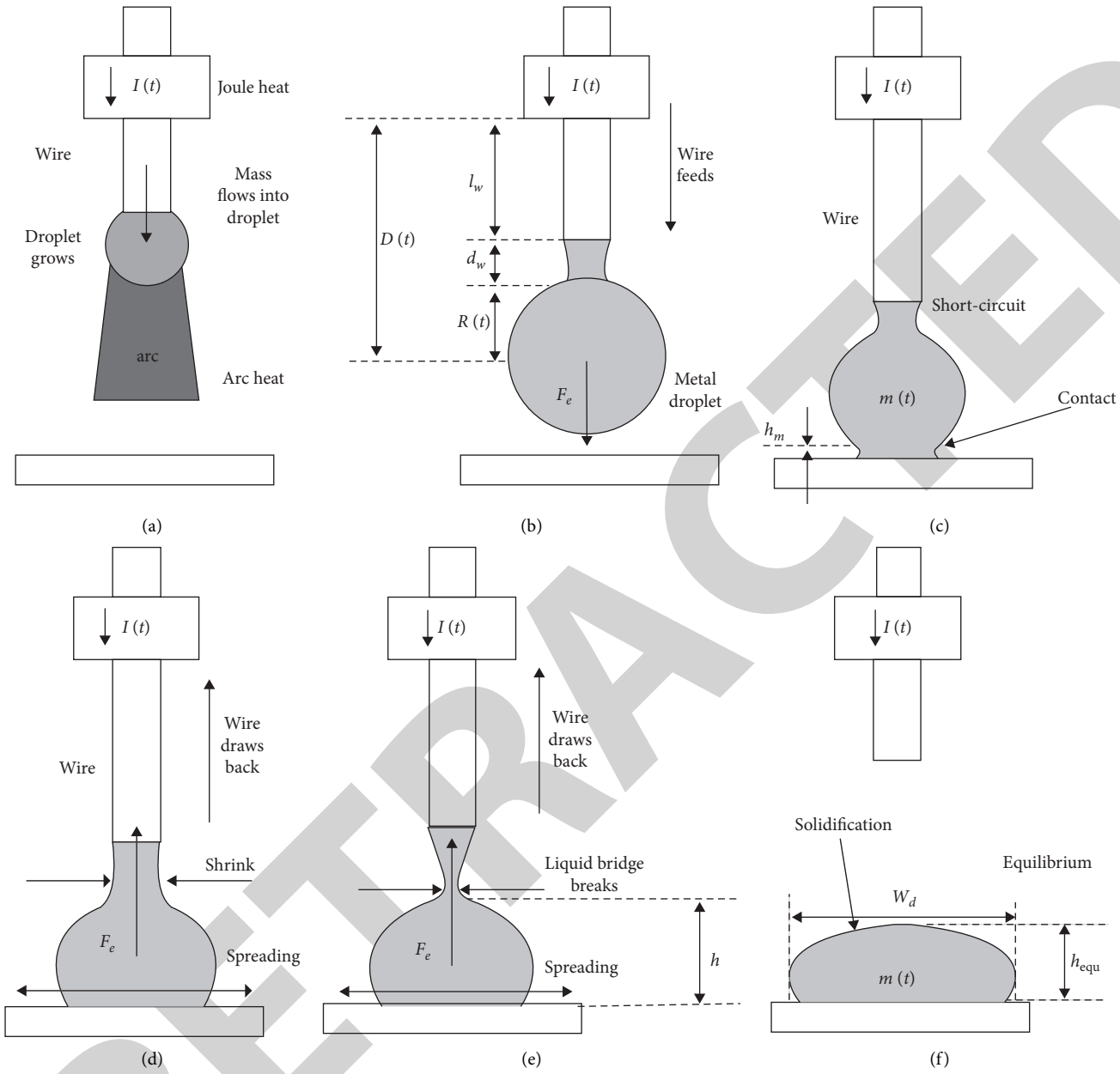


Figure 5: Short-circuit transfer model in cold metal transfer.

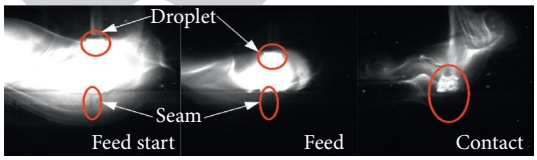


Figure 6: Electrode movement in CMT process.

In CMT, the free surface deformation of molten pool is small, the contact interface between droplet and base metal is assumed to be liquid-solid. The metal droplet is difficult to infiltrate the surface of the workpiece, the droplet cannot spread completely over the workpiece and will gradually spread to equilibrium, and the contact angle gradually changes from θ_i to θ_e, as shown in Figure 10.

According to the research by Suli et al. [26], when the droplet impacts with velocity V, the radius of the droplet can be written as

$$\left(\frac{R_B}{R_i}\right)^2 = \frac{(\mathrm{We}+12)}{3(1-\cos\theta)+4(\mathrm{We}/\sqrt{\mathrm{Ry}})}, \tag{14}$$

where $\mathrm{We} = (\rho RV^2/\gamma)$, $\mathrm{Ry} = (\rho RV/\mu)$, V is the initial velocity of the droplet when it contacts the surface of the molten pool, R_i is the initial radius of the droplet when it comes into contact with the surface of the base material, R_B

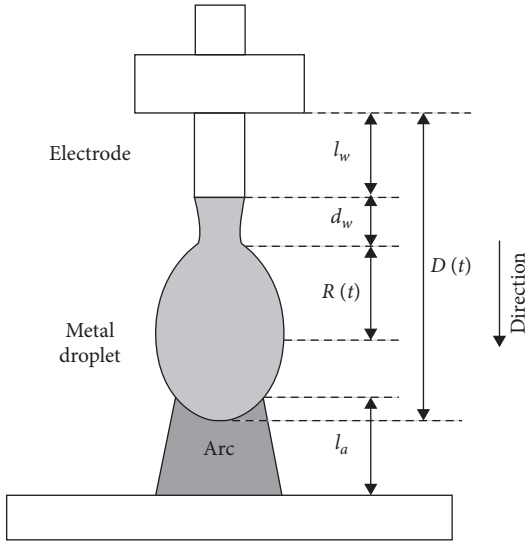


Figure 7: Distance of droplet and workpiece.

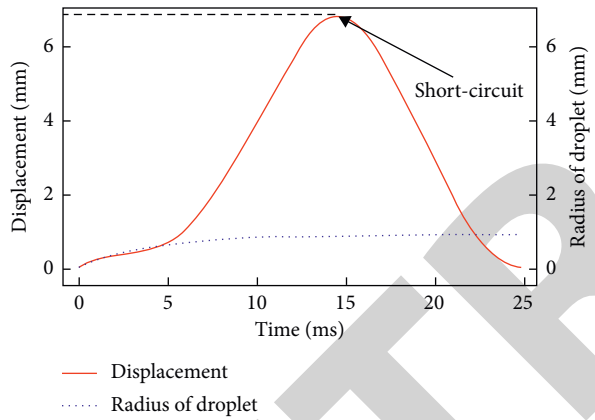


Figure 8: Motion of metal droplet in CMT (wire feed speed is 5.0 m/min).

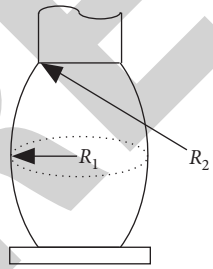


Figure 9: Molten metal bridge.

is the base radius of droplet in spreading process, γ is the surface tension coefficient, and μ is the viscosity coefficient.

Contact angle is expressed as a function of the cosine of the contact angle and time, and is used to describe spreading in reaction-limited wetting:

$$\cos\theta_e - \cos\theta = (\cos\theta_e - \cos\theta_i)\exp(-\lambda t), \quad (15)$$

where θ_i is the initial contact angle, θ_e is the contact angle in equilibrium, and λ is the kinetic constant, assuming that the kinetic constant does not change with temperature. These values are shown in Table 3.

Equations (14) and (15) lead to the following expression:

$$R_B = R_0\sqrt{\frac{(\mathrm{We}+12)}{3(1+(\cos\theta_e-\cos\theta_i)\exp(-\lambda t)-\cos\theta_e)+4(\mathrm{We}/\sqrt{\mathrm{Ry}})}}. \quad (16)$$

When the droplet spreads to the equilibrium state, the radius after solidification is the weld width because the droplet solidification time is longer than the spread time.

4. Results and Discussion

During the entire CMT welding cycle, the droplet is formed during the arcing phase, the ends of the wire are melted to form droplets, and the droplet forms a small bridge when in contact with the molten pool. At the same time, the CMT welding obtains a short-circuit signal, then cuts off the current, and the wire is pumped back to help the droplet separate from the electrode.

As shown in Figure 11, the generation of the arc and the growth of the droplet correspond to the arcing phase of the CMT, and the droplet radius increases rapidly. This stage is defined as diffusion. The initial rapid diffusion of droplet is affected by the initial velocity of droplet and the direct heating of arc. This kind of direct arc heating contributes to the rapid diffusion of droplet. At the beginning of the short-circuit transition, the formed droplets are in contact with the electrodes to the surface of the base metal, and are forced to diffuse under the action of inertial forces. This diffusion is affected by surface tension and viscous resistance, and finally reaches equilibrium. When the wire feed speed increases, the welding wire will start to move from a higher distance to the metal surface, the duration of the droplet growth process increases slightly, and the amount of wire metal melted at the end of the wire increases.

The droplet motion is recorded by a high-speed camera, and the contour of the droplet is extracted. The metal transfer and spreading of droplets recorded by the high-speed camera are shown in Figure 12.

Measurements of droplet dimensions are conducted using MATLAB (up to 10,000 frames a test). The program extracts the values of several geometrical characteristics by turning the original image into a binary image. The program is also used to find the edge and calculate the base droplet radius, as shown in Figure 13.

The comparison of the droplet spreading motion with the model calculations shows a better agreement, as shown in Figure 14. The base radius changes very rapidly at the beginning of the spreading. The initial velocity of the droplet transfer to the surface of the sheet is affected by the feed of the wire. After this rapid change, the spreading of the large droplets becomes smoother, and the welding heat is not directly heated by the arc but is conducted through the heat affect zone. According to the spreading dynamics theory, the average expansion speed of metal droplets is lower than the initial phase.

The changes in the base radius at different wire feeding speeds are shown in Figure 15. For thin-plate welding, the change in parameters causes the external force of the droplet

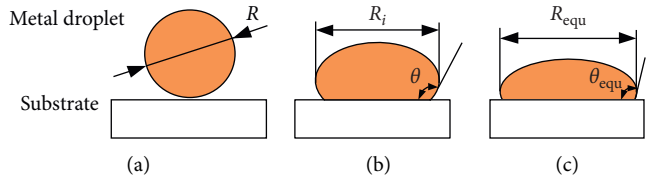


Figure 10: Metal droplet deformation.

Table 3: Physical parameter values used for calculations.

	Properties (units)	Values
θ_i	Initial contact angle (°)	40
θ_e	Contact angle in equilibrium (°)	160
λ	Kinetic constant	20

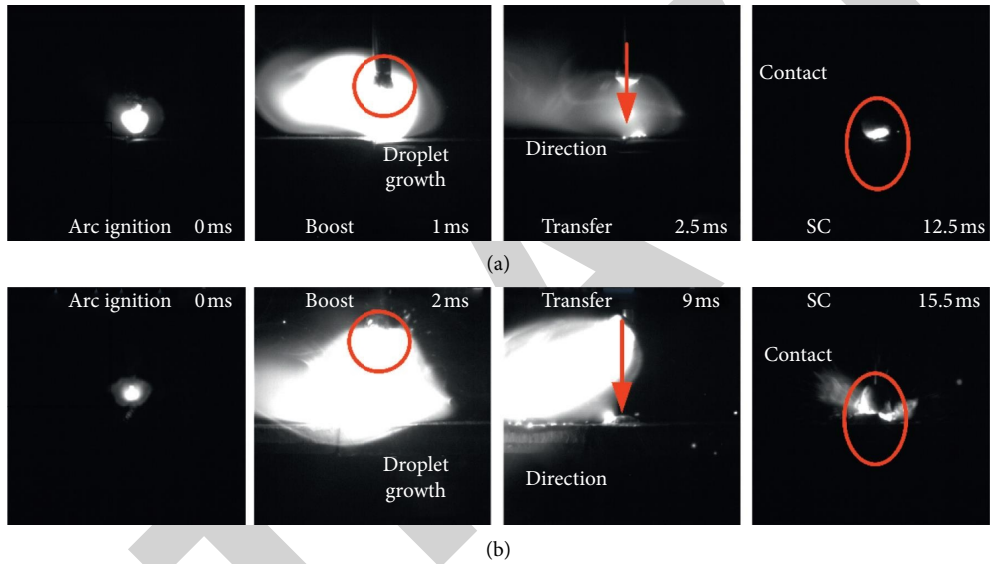


Figure 11: Arc shape and droplet transfer at different process levels: (a) wire feed speed is 3.0 m/min; (b) wire feed speed is 5.0 m/min.

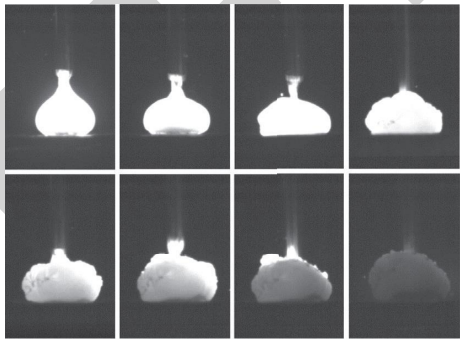


Figure 12: Droplet spreading process (wire feed speed is 5.0 m/min).

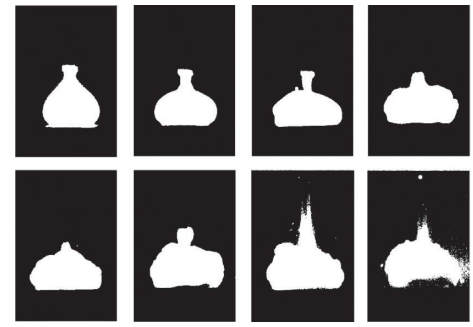


Figure 13: Binary image of the droplet radius (wire feed speed is 5.0 m/min).

to change. At the same time, the increase in the current level will change the melting rate of the wire, ensuring that the short-circuit transfer occurs smoothly at a large wire feed speed.

When the wire feeding speed is increased, the molten pool will be affected by the wire movement, thereby increasing the penetration depth and the welding width. Therefore, in the CMT process, the welding formation is mainly controlled by the movement of the electrode. The droplet transfer is mainly achieved in the form of a short-circuit transition.

The experiment also studied the movement of the droplet under different parameters to verify the versatility and guidance of the theoretical model.

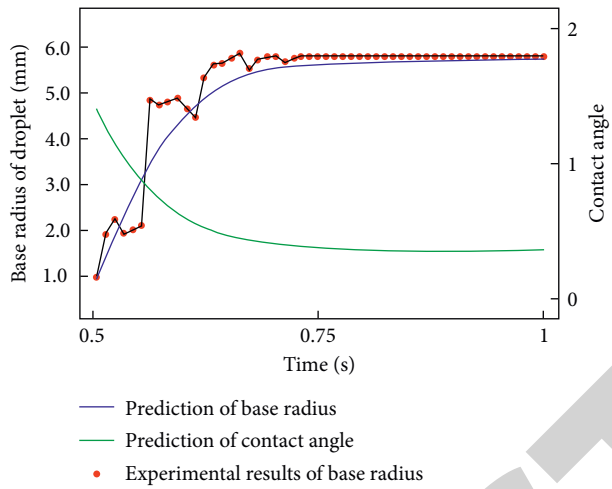


Figure 14: Variation of droplet base radius in spreading stage (wire feed speed is 5.0 m/min).

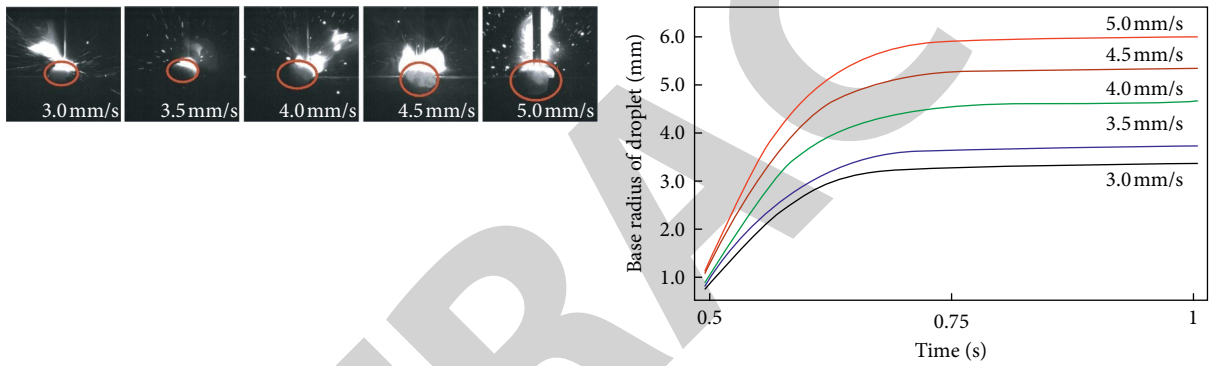


Figure 15: Predictions of the droplet spreading in different wire feed speeds.

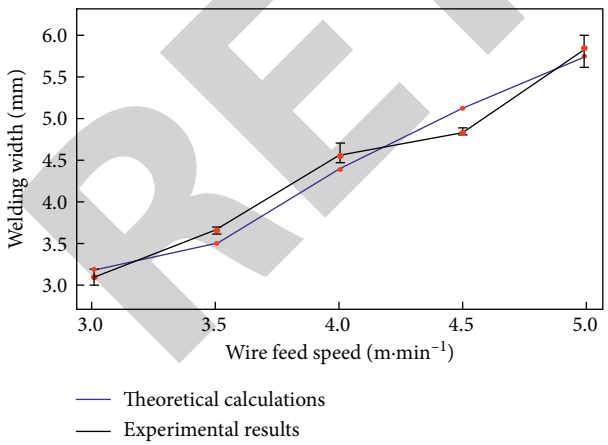


Figure 16: Welding formation in different wire feed speeds.

The model is used to calculate the welding width after droplet spreading and solidification. Figure 16 shows the welding formations in different wire feed speeds. The welding width increases with increasing wire feed speed, and the theoretical calculations show good agreement with the experimental results. In this study, the droplet spreading is considered for the base material with smooth surface, so errors between theoretical calculations and experimental results may exist.

5. Conclusions

A theoretical model of CMT is developed and used to study geometry formation of a droplet during its transfer.

According to the working mechanism of CMT, the influence of CMT pumping motion on wire feeding speed control is analyzed, and the influence of the balance between wire feeding speed and wire melting speed on wire drawing is investigated. Based on wire melting, droplet transfer, and droplet spreading, the droplet movement in CMT welding is analyzed.

Based on the analysis of CMT working mechanism and droplet transfer, the relationship between CMT pumping motion and wire melting rate is considered, the droplet growth and transition model in the CMT working environment is established. Considering the influence of the contact angle on the spreading has been considered, a spreading model of the droplet is established.

The proposed model can effectively predict the motion of droplet in CMT welding and calculate the melting width of the droplet solidified on the base metal.

The theoretical model and the experimental results show a high degree of consistency and can predict the geometry of the weld. This theory allows us to study the motion of this macroscopic droplet in the CMT process.

Abbreviations

k: Boltzmann constant
L: Electrode extension
A: Area of the electrode
Q_J: Joule heat
Q_A: Arc heat
V_a: Arc voltage
Φ: Work function of material
M_R: Wire melting rate
α: Arc heating parameter
β: Joule heating parameter
I: Welding current
l_s: Electrode extension
v_{wfd}: Wire feed speed
m_R: Droplet mass
$m(t)$: Droplet mass during droplet transfer period
R: Droplet radius in transfer process
R_i: Initial radius of the droplet when it contacts the surface of base material
R_B: Base radius of droplet in spreading process
R_e: Base radius when the droplet is spread to equilibrium
r: Electrode radius
h_m: Height of meniscus liquid surface
ρ: Mass density
F_e: External forces
μ_0: Permeability of free space
θ_z: Conduction zone radius
d: Initial distance between the welding wire and the molten pool
$D(t)$: Displacement of the droplet
l_w: Wire electrode length
d_w: Distance between the droplet and the wire electrode
l_a: Arc length
θ_i: Initial contact angle
θ_e: Contact angle in equilibrium
We: Weber number
Ry: Reynolds number
γ: Surface tension coefficient
μ: Viscosity coefficient.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the Natural Science Foundation of Shanghai (grant no. 20ZR1422600), the Open Fund of the Jiangsu Laboratory of Lake Environment Remote Sensing Technologies (grant no. JSLERS-2019-003), and the Open Project of the Shanghai Key Laboratory of Digital Manufacture for Thin-Walled Structures (grant no. 2019-002).

References

[1] B. Wang, S. J. Hu, L. Sun, and T. Freiheit, "Intelligent welding system technologies: state-of-the-art review and perspectives," *Journal of Manufacturing Systems*, vol. 56, pp. 373–391, 2020.

[2] S. B. Chen and N. Lv, "Research evolution on intelligentized technologies for arc welding process," *Journal of Manufacturing Processes*, vol. 16, no. 1, pp. 109–122, 2014.

[3] J. Feng, H. Zhang, and P. He, "The CMT short-circuiting metal transfer process and its use in thin aluminium sheets welding," *Materials & Design*, vol. 30, no. 5, pp. 1850–1852, 2009.

[4] H. Lei, Y. Li, and B. E. Carlson, "Cold metal transfer spot welding of 1 mm thick AA6061-T6," *Journal of Manufacturing Processes*, vol. 28, pp. 209–219, 2017.

[5] R. Cao, G. Yu, J. H. Chen, and P.-C. Wang, "Cold metal transfer joining aluminum alloys-to-galvanized mild steel," *Journal of Materials Processing Technology*, vol. 213, no. 10, pp. 1753–1763, 2013.

[6] R. Cao, Q. Huang, J. H. Chen, and P.-C. Wang, "Cold metal transfer spot plug welding of AA6061-T6-to-galvanized steel for automotive applications," *Journal of Alloys and Compounds*, vol. 585, pp. 622–632, 2014.

[7] P. Wang, S. Hu, J. Shen, and Y. Liang, "Characterization the contribution and limitation of the characteristic processing parameters in cold metal transfer deposition of an Al alloy," *Journal of Materials Processing Technology*, vol. 245, pp. 122–133, 2017.

[8] M. Chen, D. Zhang, and C. Wu, "Current waveform effects on CMT welding of mild steel," *Journal of Materials Processing Technology*, vol. 243, pp. 395–404, 2017.

[9] N. P. Kumar, S. A. Vendan, and N. S. Shanmugam, "Investigations on the parametric effects of cold metal transfer process on the microstructural aspects in AA6061," *Journal of Alloys and Compounds*, vol. 658, pp. 255–264, 2016.

[10] A. B. Murphy, V. Nguyen, Y. Feng, D. G. Thomas, and D. Gunasegaram, "A desktop computer model of the arc, weld pool and workpiece in metal inert gas welding," *Applied Mathematical Modelling*, vol. 44, pp. 91–106, 2017.

[11] H. Komen, M. Shigeta, and M. Tanaka, "Numerical simulation of molten metal droplet transfer and weld pool convection during gas metal arc welding using incompressible smoothed particle hydrodynamics method," *International Journal of Heat and Mass Transfer*, vol. 121, pp. 978–985, 2018.

[12] C. S. Wu, M. A. Chen, and S. K. Li, "Analysis of excited droplet oscillation and detachment in active control of metal transfer," *Computational Materials Science*, vol. 31, no. 1-2, pp. 147–154, 2004.

[13] A. R. Doodman Tipi, S. K. Hosseini Sani, and N. Pariz, "Frequency control of the drop detachment in the automatic GMAW process," *Journal of Materials Processing Technology*, vol. 216, pp. 248–259, 2015.

Hindawi
Shock and Vibration
Volume 2023, Article ID 9795858, 1 page
https://doi.org/10.1155/2023/9795858

Retraction

Retracted: Research on Quality Anomaly Recognition Method Based on Optimized Probabilistic Neural Network

Shock and Vibration

Received 17 October 2023; Accepted 17 October 2023; Published 18 October 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] L. Li, K. Chen, J. Gao, and H. Li, "Research on Quality Anomaly Recognition Method Based on Optimized Probabilistic Neural Network," *Shock and Vibration*, vol. 2020, Article ID 6694732, 11 pages, 2020.

Hindawi
Shock and Vibration
Volume 2020, Article ID 6694732, 11 pages
https://doi.org/10.1155/2020/6694732

Research Article

Research on Quality Anomaly Recognition Method Based on Optimized Probabilistic Neural Network

Li-li Li,[1,2] **Kun Chen**,[1,2] **Jian-min Gao**,[1,2] **and Hui Li**[1,2]

[1]*State Key Laboratory of Mechanical Manufacturing Systems Engineering, Xi'an Jiao Tong University, Xi'an 710049, China*
[2]*Institute of New Energy Equipment and Quality Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

Correspondence should be addressed to Li-li Li; lilili199013@stu.xjtu.edu.cn

Received 21 October 2020; Accepted 29 October 2020; Published 12 November 2020

Academic Editor: Tangbin Xia

Aiming at the problems of the lack of abnormal instances and the lag of quality anomaly discovery in quality database, this paper proposed the method of recognizing quality anomaly from the quality control chart data by probabilistic neural network (PNN) optimized by improved genetic algorithm, which made up deficiencies of SPC control charts in practical application. Principal component analysis (PCA) reduced the dimension and extracted the feature of the original data of a control chart, which reduced the training time of PNN. PNN recognized successfully both single pattern and mixed pattern of control charts because of its simple network structure and excellent recognition effect. In order to eliminate the defect of experience value, the key parameter of PNN was optimized by the improved (SGA) single-target optimization genetic algorithm, which made PNN achieve a higher rate of recognition accuracy than PNN optimized by standard genetic algorithm. Finally, the above method was validated by a simulation experiment and proved to be the most effective method compared with traditional BP neural network, single PNN, PCA-PNN without parameters optimized, and SVM optimized by particle swarm optimization algorithm.

1. Introduction

The core of quality control lies in the full utilization and analysis of quality-related data. The goal of quality control is to identify quality problems early and take appropriate measures to solve them so that the production system can remain stable for a long time. Therefore, how to effectively organize and utilize quality data has become a problem which is widely studied and focused on by enterprises and scholars. SPC (statistical process control) is the most widely used quality control method at present. Control chart was first proposed by W. A. Shewhart of the United States, and the state of a series of distribution of points, which is drawn on the control chart according to certain processing methods, is used to determine whether the production process is in a stable and controlled state. Based on the principle of saliency statistics, the control chart divides fluctuations of the production system into system fluctuations and abnormal fluctuations and considers that system fluctuation always exists and cannot be eliminated; however, abnormal fluctuations will show certain rules on the control chart and can be eliminated by certain control means [1].

Since the control chart can monitor the running state of production system and predict possible quality problems, consequently, the identification for quality anomaly is mainly about recognizing the anomaly pattern of SPC control chart. Early anomaly pattern recognition mainly adopted the rule-based ES for anomaly patterns determination [2]. In fact, the utilization of rules in ESs based on statistical properties has the difficulty that similar statistical properties may result from different patterns, and this could create problems of incorrect recognition [3]. Besides, this recognition method of anomaly pattern by establishing the expert system based on rules needs to establish a large number of determination rules, which could cause the problem of rule combination explosion; moreover, the rate of recognition accuracy was greatly affected for the existence of random factors [4]. The ANNs are capable of learning and self-organizing and hence are widely adopted in the field of control chart pattern recognition (CCPR). Aiming at the

common difficulty in existing control chart pattern recognition approaches of discrimination between different types of patterns, which share similar features, Guh et al. proposed an artificial neural network for online control chart pattern detection and discrimination [5]. Spoerre and Perry proposed the method of control chart pattern recognition using a backpropagation neural network, and the results indicated that the performance of the backpropagation neural network was very accurate in identifying control chart patterns [6]. El-Midany et al. proposed a framework for the pattern recognition of control chart in a multivariate process using artificial neural networks [7]. Wu and Wu investigated the method of wavelet neural network-based control chart patterns recognition [8]. Ebrahimzadeh et al. presented a novel hybrid intelligent method (HIM) for recognition of the common types of control chart pattern (CCP) using K-MICA clustering and neural networks [9]. Addeh et al. proposed a method of pattern recognition of control chart using RBF neural network with new training algorithm and practical features [10]. Since late 1990s, until the year of 2010, feature-based and wavelet-denoise input representation techniques have been studied for boosting the recognition performance of ANNs. The most significant works include wavelet-ANN [11], shape features-ANN [12], and statistical features-ANN [13].

Most of the previous works used raw process data as the input vector for CCPR [14]. In recent years, main research contents are concentrated on feature extraction and classification algorithm improvement. Recognizing and diagnosing abnormalities based on original data directly may lead to algorithms' training time too long because the dimension of input data is too high. Consequently, some studies first extracted these features of control chart [15] and then identified patterns of the control chart. Zhang and Cheng integrated shape and statistical features with principal component analysis (PCA) to reduce features dimensions [16]. Gauri and Chakraborty first extracted seven features of control chart based on the CART regression tree and then used the artificial neural network ANN to recognize patterns, which achieved good results [17]. Gauri and Chakraborty investigated feature-based pattern recognition of the control chart and found that the feature-based ANN recognizers could achieve better recognition performance than the feature-based heuristic recognizers [18]. Gauri and Chakraborty studied the recognition of control chart patterns by improved selection of features [17]. Ebrahimzadeh and Ranaee researched on the method of control chart pattern recognition using an optimized neural network and efficient features, and the entropies of the wavelet packets are applied for the first time in the feature extraction module [19]. Cheng et al. studied the recognition method of control chart patterns using a neural network-based pattern recognizer with features extracted from correlation analysis, and the superior performance of the feature-based recognizer over the raw data-based one is demonstrated using synthetic pattern data [20]. Xanthopoulos and Razzaghi completed pattern recognition of control chart based on the combination of statistical feature extraction and weighted support vector machine [21]. Kao et al. proposed a multistage pattern recognition scheme of control chart based on independent component analysis and support vector machine and achieved accurate and stable recognition results [22]. Zhao et al. researched the recognition of control chart pattern using improved supervised locally linear embedding and support vector machine, which effectively eliminated redundant features in the feature set and reduced the complexity of the classification model [23]. Liu and Zhou first extracted the mean characteristics of data and then divided the anomaly patterns into three categories through a layer of ANN, and the second layer realized specific classification of anomalies through three SVM classifiers with their key parameters optimized by PSO (particle swarm optimization) algorithm [24]. Zan et al. proposed a method of control chart pattern recognition based on the convolution neural network, and the feasibility and effectiveness were verified through Monte Carlo simulation [25]. Besides, some new neural networks such as Elman network [26] and spiking neural network (SNN) [27] were also applied to the identification of anomaly patterns. As can be seen from the above review that most of the scholars adopted ANNs in the field of CCPR because of their significant advantages, however, one disadvantage with ANN is the difficulty in selecting the topology and architecture of the ANN model, and the network architecture and training parameters of ANN are quite hard to determine too. In view of this problem, Guh proposed the genetic algorithm (GA) to evolve the configuration and the training parameter set of the ANN model in solving the online CCPR problem [28]. Addeh et al. investigated the design of an accurate system for the control chart patterns (CCPs) recognition, and the cuckoo optimization algorithm (COA) was applied to find the optimal parameters of the radial basis function neural network (RBFNN) [29].

In recent years, with the successful application and in-depth investigation of many machine learning techniques (e.g., artificial neural networks (ANNs) and support vector machines (SVMs)) in the field of CCPR, there are a few scholars paying attention to the identification of concurrent CCPs (two or more basic CCPs occurring simultaneously). Yang et al. finished the identification and quantification of concurrent control chart patterns using extreme-point symmetric mode decomposition and extreme learning machines [30]. Zhang et al. investigated the recognition of mixture control chart patterns based on fusion feature reduction and fireworks algorithm-optimized MSVM [31]. However, the current study about the identification of concurrent CCPs is not far enough and needs to be studied further.

Although all the above studies provided very critical references for the sequent research work, all these methods currently have certain defects. The topology and architecture of the ANN model are very difficult to select, and the network architecture and training parameters of ANN are quite hard to determine too. SVM is a two classifier, which is not enough efficient in the problem of multiclassification

identification of abnormal patterns. In addition, in most cases, there are always several kinds of abnormal patterns appearing simultaneously in actual production, and current research mostly focuses on the identification of single anomalies. Consequently, the research on CCPR needs to be further expanded.

Based on the above explanation, this paper proposes a kind of quality anomaly recognition method based on a probabilistic neural network optimized by improved genetic algorithm, which finished the recognition of mixed anomaly patterns. As can be seen from the simulation experiment, the proposed method has certain advantages in the accuracy of anomaly recognition, the speed of network training, and the ease of method.

2. Quality Abnormal Patterns and Description

In 1958, Western Electric Company first proposed the problem of control chart pattern recognition and summarized abnormal patterns of the control chart [32], and accordingly, later scholars made in-depth research on abnormal pattern of the control chart and identification problem. With the development of artificial intelligence technology, using artificial intelligence technology to recognize and determinate the control chart abnormal pattern has become one of the important research contents in the field of quality diagnosis. There are six basic modes of the control chart, as shown in Figure 1. The combination of different anomaly modes and abnormal modes reflects different states and unstable factors of production process. Starting from the abnormal mode of the control chart, it can discover hidden quality dangers in the current production process and then take corresponding solutions timely.

The production process can be expressed by the following formula:

$$x(t) = \mu + d(t) + r(t), \quad (1)$$

where t represents the sampling moment of production data, $x(t)$ represents the observation value of production process, μ is the mean of quality characteristics when production is stable, $d(t)$ is the abnormal fluctuations caused by special factors, and different $d(t)$ leads to different abnormal patterns in the production process. Under normal circumstances, $d(t)$ is 0; $r(t)$ is normal fluctuations caused by random factors, which obey normal distribution with a mean of 0 and a variance of σ^2.

Under normal circumstances, $d(t) = 0$, the control chart is in a normal mode, and the above points are randomly distributed; $d(t) = A \times \sin(2\pi t/T)$ is in the periodic mode, and A and T are amplitude and period, respectively; in the up (down) trend mode $d(t) = \pm k \times t$, in which k is the slope of the trend, positive when up and negative when down; in the up (down) step mode $d(t) = \pm 1(t - t_0) \times s$, in which t_0 is the moment when the step mode occurs and s is step amplitude; when $t \geq t_0$, $1(t - t_0) = 1$ and otherwise 0. For the mixed anomaly mode, it can be described as the addition of abnormal fluctuation $d(t)$ of basic anomaly mode.

3. Quality Anomaly Recognition Model Based on Optimized PNN

The model of quality anomaly recognition based on optimized probabilistic neural network (PNN) was established and is shown in Figure 2, including the three stages: feature extraction, pattern classification, and parameter optimization. In the stage of feature extraction, the dimension of original sample data is reduced to certain dimension by PCA, and 85% information of the original data is retained. The PNN is used to identify the control pattern, including basic pattern and mixed pattern recognition. The improved single-objective optimization genetic algorithm SGA is used to optimize main parameters of the PNN network, and the recognition accuracy of training samples is used as the fitness function of the SGA.

The detailed steps of the model are as follows:

Step 1: the generation of sample data. The Monte Carlo method simulated and generated sample data of six basic modes and the eight mixed modes.

Step 2: feature extraction. As the input of PNN, the sample data were decreased to certain dimension by PCA.

Step 3: establishment of PNN network. Determine the network structure and set parameters range for subsequent pattern recognition and algorithm optimization.

Step 4: optimization of parameters. The improved SGA was used to optimize parameters. Recognition accuracy of PNN is taken as the fitness function to find the optimal smoothing coefficient σ.

Step 5: classification. The PNN network after parameter optimization realized the recognition of control pattern abnormal mode.

3.1. Feature Extraction and Dimensionality Reduction of Control Chart Based on PCA. In this paper, the feature extraction method based on PCA is a new feasible data processing method of the control graph mode in addition to the calculation of shape and statistical features and wavelet transform. PCA eliminates the complexity of calculating multiple features and also does not need to select the appropriate number of decomposition layers and wavelet levels as the wavelet transform. It is certificated by experiments that the same processing effect can be achieved.

Principal component analysis (PCA) is the most commonly used method of feature dimension reduction in data processing. It can extract the main components from complex multidimensional data and greatly improve the training speed of model without losing too much model quality [33]. The main method of PCA is to find a set of mutually orthogonal coordinate axes from the distribution space of original data. The first new coordinate axis is the direction with the largest variance in the original data, and the second new coordinate axis is the direction with the largest variance in orthogonal plane with the first coordinate

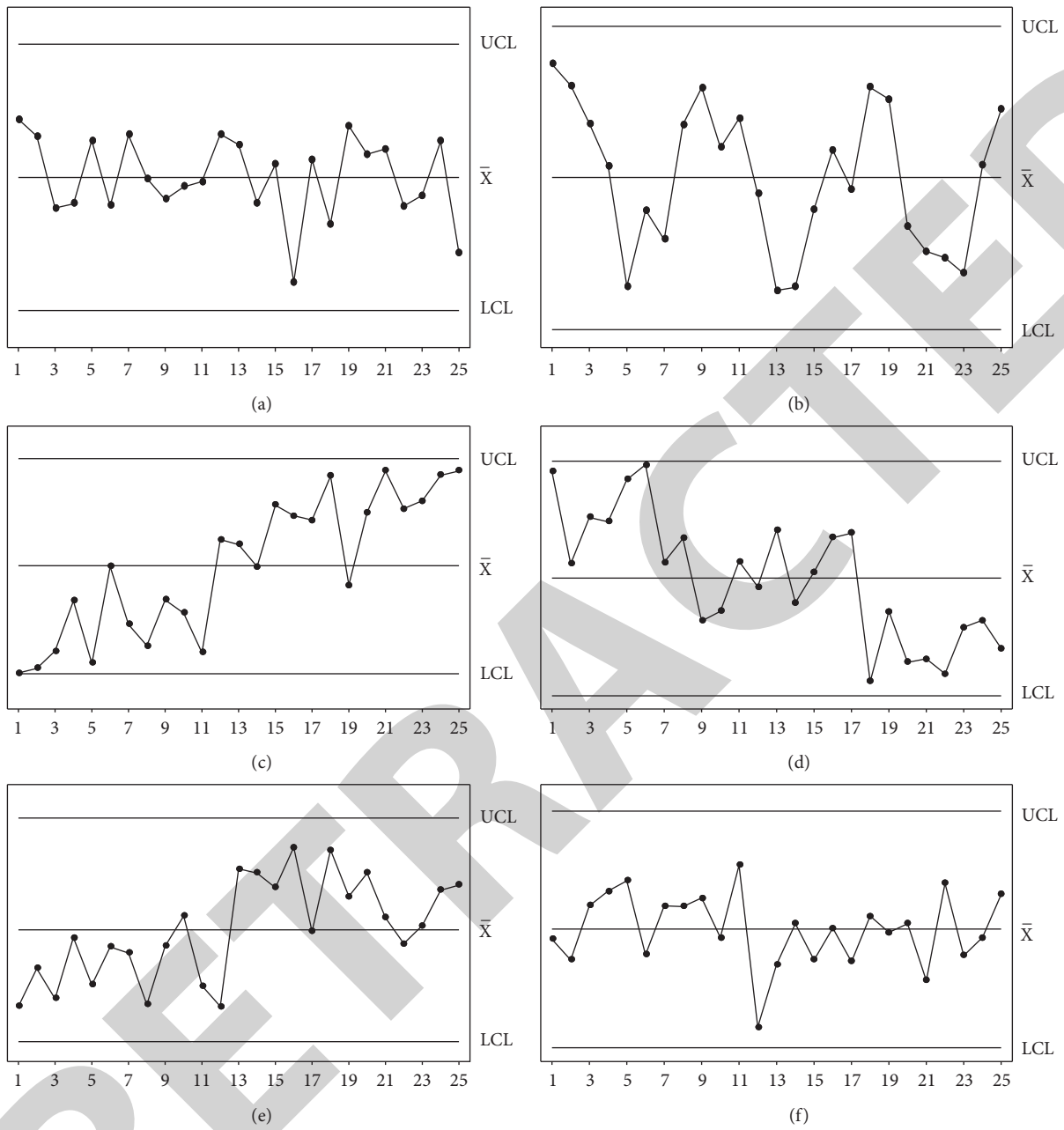


Figure 1: Basic quality anomaly modes: (a) normal mode; (b) cycle mode; (c) upward trend mode; (d) downward trend mode; (e) upward step mode; (f) downward step mode.

axis, and so on; a series of such axes can be obtained. In these axes, the first few axes containing most of the variances are used to achieve dimensionality reduction of data features.

Figures 3–6 show the two-dimensional and three-dimensional scatter plots of original data and PCA processed data, respectively. It can be clearly seen from figures that data of different anomaly types are obviously distinguished after PCA principal component analysis, which is convenient for subsequent classification identification of PNN.

3.2. Anomaly Recognition Method Based on PNN. Probabilistic neural network (PNN) is one kind of radial basis neural network, and it replaces the commonly used *S*-shaped activation function in neural networks with exponential function to calculate the nonlinearity discrimination boundary, which is close to the Bayesian optimal decision surface [34]. PNN is a kind of nonparametric estimation method based on the Bayesian optimal classification decision theory and probability density function in statistics [35], unlike the backpropagation algorithm used by traditional neural networks, and PNN is a kind of forward propagation algorithm without feedback. The structure of a typical probabilistic neural network is shown in Figure 7.

PNN is composed of the input layer, the sample layer, the summation layer, and the output layer (the competition layer). The number of neurons in the input layer is the

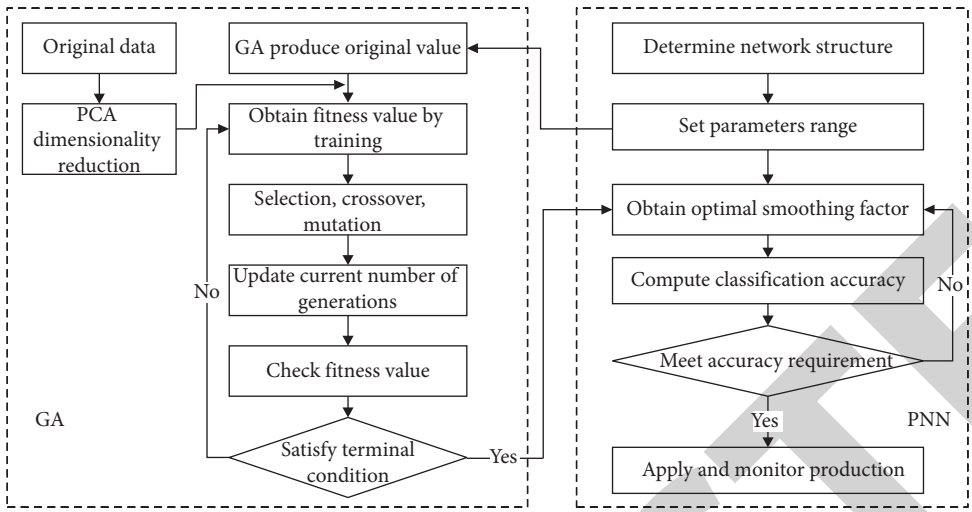


Figure 2: Quality anomaly recognition model.

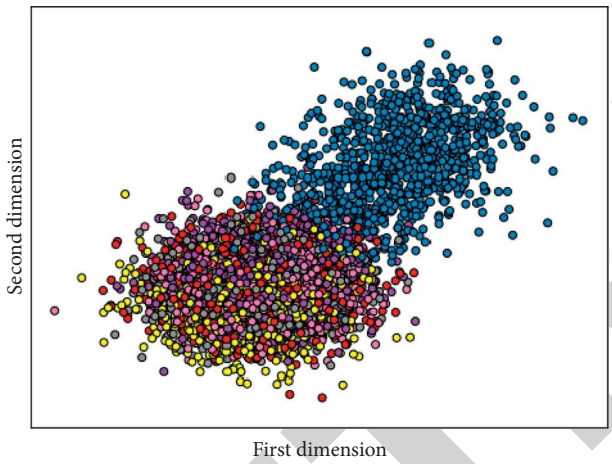


Figure 3: Two-dimensional scatter plot of original data.

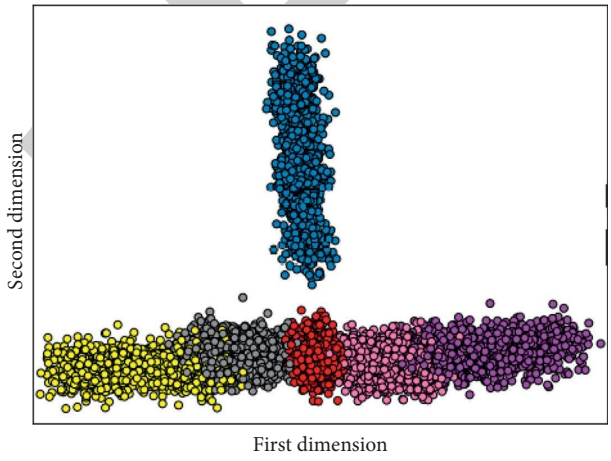


Figure 4: Two-dimensional scatter plot of PCA processed data.

number of dimensions of the feature vector, which is the data dimension after PCA processing, and the input layer calculates the distance between the input vector and all the

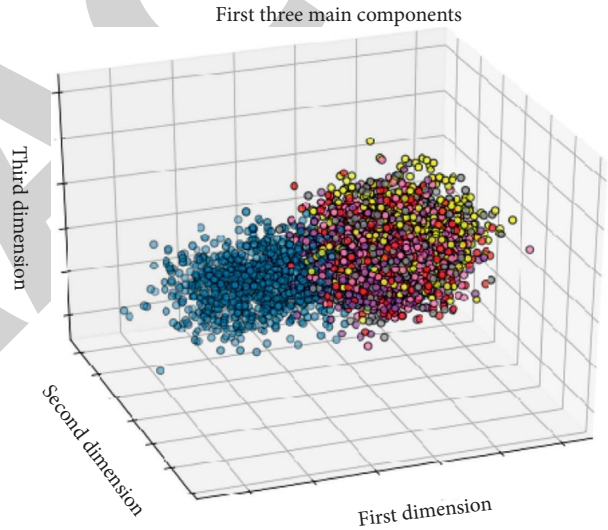


Figure 5: 3D scatter plot of original data.

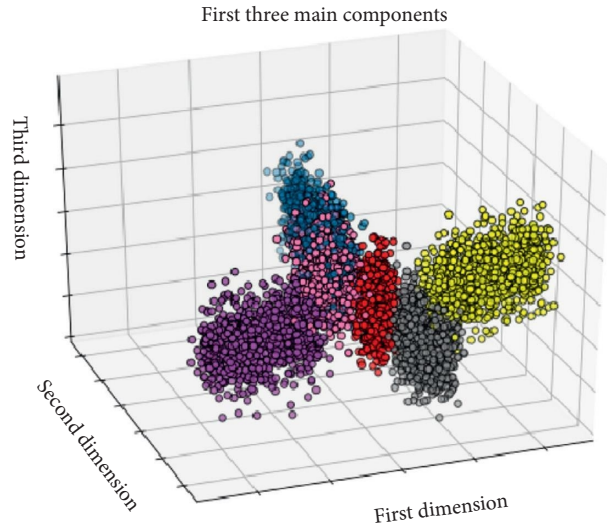


Figure 6: 3D scatter plot of data after PCA processing.

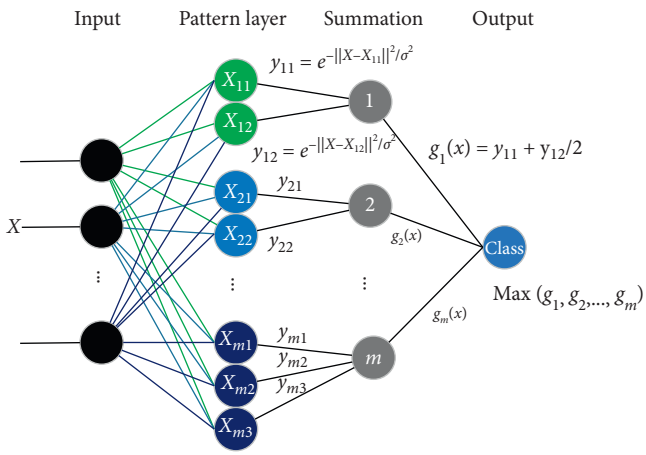


Figure 7: Probabilistic neural network structure.

training sample vectors. The sample layer includes all the training samples, the number of neurons is the number of all training samples, and the activation function is a Gaussian function. The number of neurons in the summation layer is the number of categories, and the output of sample layer is added by classification. The output layer is also called competition layer, which has only one neuron, and the kind of output with the highest probability value is 1.

In this paper, the input layer of PNN network for anomaly recognition is the control graph data after PCA processing. The number of neurons in the input layer is determined by the result of PCA processing. The number of neurons in the sample layer is the number of training samples. The number of neurons in the summation layer is 10, which is 10 categories representing anomalies, including six basic modes and four mixed modes. Compared to the traditional BP neural network and RBF neural network, the learning process of PNN network is simple, its learning speed is fast, classification is more accurate, and its tolerance for errors and noise is also high without local minimum value problem; when the number of representative training samples increases large enough, the classifier must be able to achieve Bayesian optimality.

3.3. Optimization of PNN Network Parameters Based on Genetic Algorithm. In the PNN neural network model, the only parameter that needs to be adjusted is σ, which is also called the smoothing parameter. When σ is too small, it only acts as an isolation for individually trained samples, and it is essentially the nearest neighbor classifier; when σ is too large, it cannot distinguish the details completely and may not achieve the ideal classification effect for different categories with inconspicuous boundaries, and the effect is close to the linear classifier.

Genetic algorithm (GA) [36] is a computational model that simulates natural selection of Darwin's biological evolution theory and biological evolution process of genetic mechanisms. It is a method to search optimal solutions by simulating natural evolutionary processes. Based on the principle of biological evolution, genetic algorithm generates new groups by selection, crossover, and mutation continuously according to individual fitness value in each generation group, which enables the population to evolve continuously. At the same time, the global parallel search technology is used to search and optimize optimal individuals for finding approximate optimal solution for the problems. The genetic algorithm is not limited to the continuity and differential of functions, and the result is globally optimal. Therefore, genetic algorithm is used to optimize smoothing coefficient of the PNN probabilistic neural network for finding optimal parameters. The specific process is as follows:

(1) The range of smoothing factor σ is set, and the initial population $\{\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_M\}$ is randomly generated, where M is the scale of population and current number of generation $t = 1$.

(2) According to the smoothing factor obtained by chromosome, the PNN network is constructed, the number of correct classifications and the accuracy rate is calculated, and the fitness function of the chromosome is calculated accordingly.

(3) Winners are selected, and operations such as crossover and mutation are performed to obtain the next generation population.

(4) Current generations $t = t + 1$ are updated.

(5) It is checked whether termination condition is met; that is, the given number of iterations is reached. If the end condition is met, the algorithm stops running; otherwise, it returns to step 2.

(6) The individual with the best fitness value is used as the result of algorithm optimization and input PNN network for obtaining the final recognition model.

4. Simulation Experiment Analysis

4.1. Data Simulation and Parameter Setting. The Monte Carlo method is used to generate 1000 samples, respectively, for six basic patterns and four mixed patterns, which is a total of 10,000 samples of control pattern. 70% of samples as a training set were chosen and 30% as a test set. Each of the control chart samples includes 25 data points, which are set according to the control chart standard of the SPC manual in TS16949 series quality technology standard of American Automotive Industry Action Group (AIAG). The parameters values of each abnormal mode are shown in Table 1.

4.2. Feature Extraction and Dimensionality Reduction. Using PCA to reduce dimensionality and extract feature for original data, generally take the dimension of the former 85% of original data, which is exactly the sum of data of the former nine dimensions. Therefore, take the principal component with 85% variance contribution rates as the data after reducing dimension. Figure 8 shows the principal components with cumulative variance contribution rate of 85% and their respective proportions.

According to the results of principal component analysis (PCA), the original data in 25 dimensions are reduced to

Table 1: Parameter settings for various abnormal conditions.

Control chart mode	Mode parameters	Parameters values
Normal mode (NOR)	Mean value μ, standard deviation σ	$\mu = 0, \sigma = 1$
Cycle mode (CYC)	Cycle T, amplitude A	$T \sim U(8, 16), A \sim U(1.5, 2.5)$
Upward (downward) trend mode (IT)	Slope k	$k \sim U(0.05, 0.10)$
Upward (downward) step mode (US)	Step amplitude s, time of occurrence t_0	$s \sim U(1.5, 2.5), t_0 = 2$
Cycle + trend mode (CT)	T, A, k	Value is the same as above
Cycle + step mode (CS)	T, A, s, t_0	Value is the same as above
Trend + step mode (TS)	k, s, t_0	Value is the same as above
Cycle + trend + step mode (CTS)	A, T, k, s, t_0	Value is the same as above

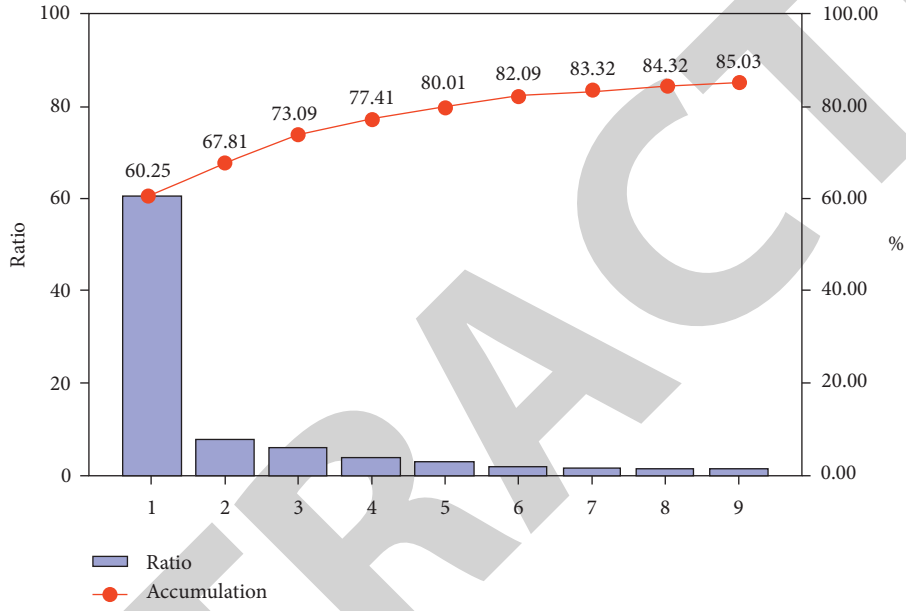


Figure 8: Principal components and their proportions.

only 9 dimensions of data. The 9 principal components are used as the dimension-reduced data and input PNN network for learning training of pattern recognition, which can greatly reduce the training time of the model and increase the accuracy of recognition result.

4.3. Algorithm Parameters Optimization. Use the improved single-objective optimization genetic algorithm (SGA) instead of standard genetic algorithm flow to generate offspring by crossover and mutation at first, then the two generations of father and child are merged, and finally, select and retain elite population from merged population, which can avoid that the best individual will be destroyed by the hybridization operation.

The optimization algorithm has 100 individuals in per generation population and evolves 200 generations in total. When SGA is running, a set of smoothing parameters are randomly generated, generating the offspring by crossover and mutation, and then the parents and the progeny are combined and select individuals with high fitness to survive. The final optimization results of improved SGA are shown in Figure 9; blue curve represents the rate of average training accuracy of each generation population, and orange curve represents the optimal rate of recognition accuracy of each generation population. When the algorithm runs to the 11th generation, the rate of training accuracy of the best individual in the population reaches the highest with the optimal smoothing parameter of 1.7385 and the optimal objective function value of 95.10%, which is also the rate of training accuracy. However, the optimal smoothing parameter is 0.8161 when PNN is optimized by traditional standard genetic algorithm, and the rate of recognition accuracy of PNN is only 81.69%; the optimization results of PNN optimized by standard genetic algorithm is shown in Figure 10. Consequently, PNN optimized by the improved SGA has a higher rate of recognition accuracy than that of standard genetic algorithm.

4.4. Model Test Result. Put optimized parameters into the model and input test sample data, and then, the final recognition results are shown in Table 2.

It can be seen from the model test results that the proposed method in this paper has a higher recognition accuracy for the basic model, most recognition accuracy is above 97%, and the average recognition rate for the mixed mode is also more than 90%. However, it also found that, in

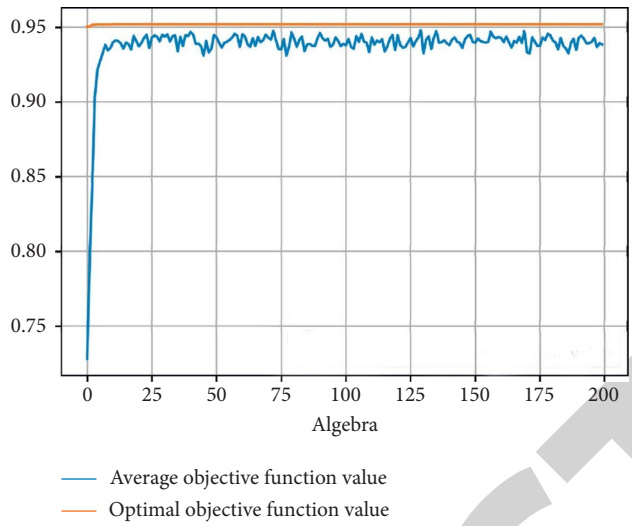


Figure 9: Improved genetic algorithm optimization results.

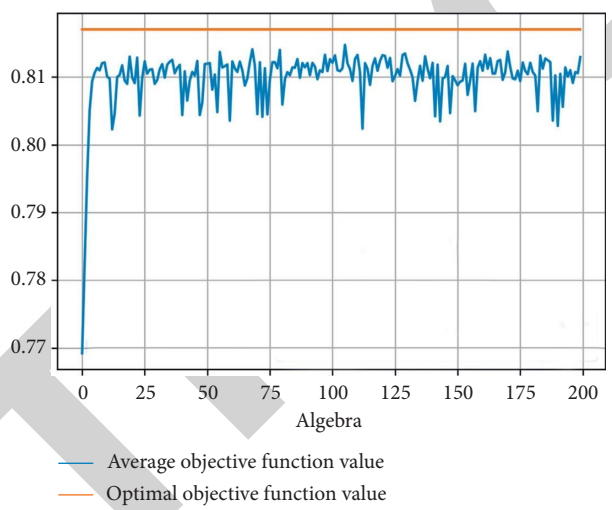


Figure 10: Standard genetic algorithm optimization results.

the cyclic step mode, the cyclic trend step mode, and the cyclic trend mode, where the recognition accuracy is relatively low, a considerable part of modes is recognized as other modes, which indicates that the distinction of modes for trend components and steps components of control chart needs to be further improved. In the future, it may be considered to take other measures to classify and recognize these two modes, such as taking different smoothing parameters for different modes, which can make the model achieve a better recognition effect.

4.5. Comparison with Other Methods. In order to compare with the PCA-SGA-PNN method proposed in this paper, the BP, PNN, and PCA-PNN control chart recognition models are established, respectively. Among them, (1) the number of input layer neurons of BP neural network is 25 and the number of hidden layers is 3. (2) Use PNN network for pattern recognition without data processing. (3) The data after PCA dimensionality reduction is input into the PNN network for identification, and no parameter optimization is performed. (4) The PNN parameter is optimized by the improved SGA method, and then, PNN is used to identify the control chart model, which is the method proposed in this paper. (5) A recognition model is established by SVM optimized by particle swarm optimization algorithm. The comparison results of several methods are shown in Table 3.

From the analysis of the results in the table, the following conclusions can be drawn:

(1) The recognition accuracy of a single BP neural network or PNN network is relatively low, and the recognition accuracy of the model with PCA analysis and dimensionality reduction is obviously improved, indicating that feature extraction is beneficial to the recognition of basic modes and mixed modes of control graphs.

(2) The recognition accuracy of the model after optimizing PNN network parameters by SGA is further improved. Through parameters optimization, the

Table 2: Test results of quality abnormality identification model.

Classification methods	Recognition rate (%)	Actual categories	Pattern types identified (%)									
			NOR	CYC	UT	DT	US	DS	CT	CS	TS	CTS
PCA-SGA-PNN	96.0	NOR	99.1	0	0.6	0.3	0	0	0	0	0	0
		CYC	1.0	97.7	0	0	0	0	1.3	0	0	0
		UT	1.0	0	97.6	0	1.4	0	0	0	0	0
		DT	1.7	0	0	97.3	0	1.0	0	0	0	0
		US	0	0	1.7	0	96.6	0	0	0	1.7	0
		DS	0	0	0	1.3	0	98.7	0	0	0	0
		CT	0	2.9	1.3	0	0	0	94.2	1.6	0	0
		CS	0	0	0	0	1.4	0	3.9	90.8	0	3.9
		TS	0	0	0	0	4.0	0	0	0	96.0	0
		CTS	0	0	0	0	0.3	0	0	7.6	0.3	91.8

Table 3: Comparison of different methods.

Recognition methods	Average recognition rate (%)	Types of control chart mode (%)									
		NOR	CYC	UT	DT	US	DS	CT	CS	TS	CTS
BP	87.4	64.9	89.6	99.3	96.0	74.4	99.4	78.7	86.6	95.2	89.8
PNN	90.8	92.5	95.1	94.9	93.4	86.5	97.7	90.0	80.9	91.6	85.5
PCA-PNN	93.5	97.3	97.0	97.3	96.4	91.9	96.8	91.6	85.4	92.0	89.1
PCA-SGA-PNN	96.0	99.1	97.7	97.6	97.3	96.6	98.7	94.2	90.8	96.0	91.8
PCA-PSO-SVM	92.3	91.2	87.7	95.3	95.5	89.3	96.1	94.5	93.1	94.5	85.6

blindness of relying on the empirical value is avoided, and a better recognition rate is obtained.

(3) Compared with SVM optimized by particle swarm optimization, the recognition accuracy of this method proposed in this paper is higher, which indicates that PNN network has certain advantages in the pattern recognition of the control graph; it is a feasible and effective identification method.

5. Conclusions

This paper proposes the method of quality anomaly pattern recognition based on probabilistic neural network optimized by improved single-target optimization genetic algorithm.

(1) In order to reduce the training time of the model, PCA (principal component analysis) was adopted in this paper and reduced the dimensionality of the original data of the control chart.

(2) PNN completed the recognition of single mode and mixed mode of control chart because of its merits of simple structure and good recognition effect. The smoothing parameter of PNN was optimized by the improved (SGA) single-target optimization genetic algorithm, which eliminated the deficiency of experience value. The improved SGA got the optimal smoothing parameter of 1.7385, which made PNN achieve the optimal objective function value of 95.10%, which is also the rate of recognition accuracy. The rate of recognition accuracy was only 81.69% when compared with PNN optimized by traditional genetic algorithm. Therefore, PNN optimized by improved SGA could achieve a higher recognition accuracy than that of traditional genetic algorithm.

(3) The above research content was verified by simulation experiments, and compared with the traditional BP neural network, single PNN, PCA-PNN model without parameter optimization, and the SVM model optimized by particle swarm optimization (PSO) algorithm, the PNN optimized by improved SGA achieved the best recognition effectiveness. Consequently, the method proposed in this paper has certain positive significance for promoting the application of control charts in actual production and helping enterprises to find out the quality anomalies timely in the production process.

Data Availability

The data used to support the findings of this study are available from the corresponding author on request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

The authors gratefully acknowledge the financial support by the National Key Research and Development Program of China (Grant no. 2019YFB1703800).

References

[1] Z. He, *Six Sigma Management*, China Renmin University Press, Beijing, China, 3rd edition, 2014.

[2] J. R. Evans and W. M. Lindsay, "A framework for expert system development in statistical quality control," *Computers & Industrial Engineering*, vol. 14, no. 3, pp. 335–343, 1988.
[3] W. Hachicha and A. Ghorbel, "A survey of control-chart pattern-recognition literature (1991–2010) based on a new conceptual classification scheme," *Computers & Industrial Engineering*, vol. 63, no. 1, pp. 204–222, 2012.
[4] Q. H. Le, L. Teng, and M. Q. Zhu, "On-line intelligent diagnosis and analysis system for quality control chart," *Computer Integrated Manufacturing System*, no. 12, pp. 1583–1587, 2004.
[5] R.-S. Guh, F. Zorriassatine, J. D. T. Tannock, and C. O'Brien, "On-line control chart pattern detection and discrimination-a neural network approach," *Artificial Intelligence in Engineering*, vol. 13, no. 4, pp. 413–425, 1999.
[6] J. K. Spoerre and M. B. Perry, "Control chart pattern recognition using a back propagation neural network," *Intelligent Systems in Design & Manufacturing III*, International Society for Optics and Photonics, Bellingham, WA, USA, 2000.
[7] T. T. El-Midany, M. A. El-Baz, and M. S. Abd-Elwahed, "A proposed framework for control chart pattern recognition in multivariate process using artificial neural networks," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1035–1042, 2010.
[8] S. Wu and B. Wu, "Wavelet neural network-based control chart patterns recognition," in *Proceedings of the 2006 World Congress on Intelligent Control & Automation*, Dalian, China, 2006.
[9] A. Ebrahimzadeh, J. Addeh, and Z. Rahmani, "Control chart pattern recognition using K-MICA clustering and neural networks," *ISA Transactions*, vol. 51, no. 1, pp. 111–119, 2012.
[10] A. Addeh, A. Khormali, and N. A. Golilarz, "Control chart pattern recognition using RBF neural network with new training algorithm and practical features," *ISA Transactions*, vol. 79, pp. 202–216, 2018.
[11] Y. Al-Assaf, "Recognition of control chart patterns using multi-resolution wavelets analysis and neural networks," *Computers & Industrial Engineering*, vol. 47, no. 1, pp. 17–29, 2004.
[12] S. K. Gauri and S. Chakraborty, "Improved recognition of control chart patterns using artificial neural networks," *International Journal of Advanced Manufacturing Technology*, vol. 36, no. 11-12, pp. 1191–1201, 2008.
[13] A. Hassan, M. S. N. Baksh, A. M. Shaharoun, and H. Jamaluddin, "Improved SPC chart pattern recognition using statistical features," *International Journal of Production Research*, vol. 41, no. 7, pp. 1587–1603, 2003.
[14] C. S. Cheng, H. P. Cheng, and K. K. Huang, "A support vector machine-based pattern recognizer using selected features for control chart patterns analysis," in *Proceedings of the 2010 IEEE International Conference on Industrial Engineering & Engineering Management*, Macao, China, 2010.
[15] V. Ranaee, A. Ebrahimzadeh, and R. Ghaderi, "Application of the PSO-SVM model for recognition of control chart patterns," *ISA Transactions*, vol. 49, no. 4, pp. 577–586, 2010.
[16] M. Zhang and W. Cheng, "Recognition of mixture control chart pattern using multiclass support vector machine and genetic algorithm based on statistical and shape features," *Mathematical Problems in Engineering*, vol. 2015, Article ID 382395, 10 pages, 2015.
[17] S. K. Gauri and S. Chakraborty, "Recognition of control chart patterns using improved selection of features," *Computers & Industrial Engineering*, vol. 56, no. 4, pp. 1577–1588, 2009.
[18] S. K. Gauri and S. Chakraborty, "Feature-based recognition of control chart patterns," *Computers & Industrial Engineering*, vol. 51, no. 4, pp. 726–742, 2006.
[19] A. Ebrahimzadeh and V. Ranaee, "Control chart pattern recognition using an optimized neural network and efficient features," *ISA Transactions*, vol. 49, no. 3, pp. 387–393, 2010.
[20] C.-S. Cheng, K.-K. Huang, and P.-W. Chen, "Recognition of control chart patterns using a neural network-based pattern recognizer with features extracted from correlation analysis," *Pattern Analysis and Applications*, vol. 18, no. 1, pp. 75–86, 2015.
[21] P. Xanthopoulos and T. Razzaghi, "A weighted support vector machine method for control chart pattern recognition," *Computers & Industrial Engineering*, vol. 70, pp. 134–149, 2014.
[22] L.-J. Kao, T.-S. Lee, and C.-J. Lu, "A multi-stage control chart pattern recognition scheme based on independent component analysis and support vector machine," *Journal of Intelligent Manufacturing*, vol. 27, no. 3, pp. 653–664, 2016.
[23] C. Zhao, C. Wang, L. Hua, X. Liu, Y. Zhang, and H. Hu, "Recognition of control chart pattern using improved supervised locally linear embedding and support vector machine," *Procedia Engineering*, vol. 174, pp. 281–288, 2017.
[24] Y. M. Liu and H. F. Zhou, "ANN-SVM identification mode and simulation analysis of quality anomaly mode in dynamic process," *Statistics and Decision*, no. 9, pp. 35–38, 2016.
[25] T. Zan, Z. Liu, H. Wang, M. Wang, and X. Gao, "Control chart pattern recognition using the convolutional neural network," *Journal of Intelligent Manufacturing*, vol. 31, pp. 703–716, 2019.
[26] X. Jiang, S. J. Wang, and K. Zhao, "Intelligent process quality control system for networked manufacturing," *Journal of Mechanical Engineering*, vol. 46, no. 4, pp. 186–194, 2010.
[27] M. H. A. Awadalla, M. Abdellatif Sadek, and M. A. Sadek, "Spiking neural network-based control chart pattern recognition," *Alexandria Engineering Journal*, vol. 51, no. 1, pp. 27–35, 2012.
[28] R.-S. Guh, "Optimizing feedforward neural networks for control chart pattern recognition through genetic algorithms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 2, pp. 75–99, 2004.
[29] J. Addeh, A. Ebrahimzadeh, M. Azarbad, and V. Ranaee, "Statistical process control using optimized neural networks: a case study," *ISA Transactions*, vol. 53, no. 5, pp. 1489–1499, 2014.
[30] W.-A. Yang, W. Zhou, W. Liao, and Y. Guo, "Identification and quantification of concurrent control chart patterns using extreme-point symmetric mode decomposition and extreme learning machines," *Neurocomputing*, vol. 147, pp. 260–270, 2015.
[31] M. Zhang, Y. Yuan, R. Wang, and W. Cheng, "Recognition of mixture control chart patterns based on fusion feature reduction and fireworks algorithm-optimized MSVM," *Pattern Analysis and Applications*, vol. 23, pp. 15–26, 2018.
[32] R. S. Guh, "Robustness of the neural network based control chart pattern recognition system to non-normality," *International Journal of Quality & Reliability Management*, vol. 19, no. 1, pp. 97–112, 2002.
[33] G. N. Costache, P. Corcoran, and P. Puslecki, "Combining PCA-based datasets without retraining of the basis vector set," *Pattern Recognition Letters*, vol. 30, no. 16, pp. 1441–1447, 2009.
[34] Y. Situ, H. Gao, D. Li et al., "Screw pairs condition monitoring based on integrated radial basis probabilistic neural