electronics

Article

Dynamic PSO-Optimized XGBoost–RFE with Cross-Domain Hierarchical Transfer: A Small-Sample Feature Selection Approach for Equipment Health Management

Yao Lei, Jianyin Zhao *, Weimin Lv and Youwei Hu

Naval Aviation University, Yantai 264001, China; leylaayy21@outlook.com (Y.L.); 15699190881@163.com (W.L.); 15066896663@163.com (Y.H.)
* Correspondence: ley11888@163.com

Abstract

In equipment health management, inefficient key feature selection and model overfitting caused by data scarcity in small-sample scenarios severely restrict the practical applications of predictive maintenance technologies. To address this challenge, this study proposes an improved key feature selection method integrating dynamic particle swarm optimization (PSO) and cross-domain transfer learning. First, principal component analysis (PCA) is employed for the dimensionality reduction of high-dimensional health-related features. An improved PSO algorithm is then used to dynamically optimize XGBoost hyperparameters, coupled with a recursive feature elimination (RFE) framework to screen for key features. A hierarchical transfer strategy is then introduced to address small-sample data limitations in the target domain via source domain knowledge transfer, achieving cross-domain feature space alignment and model parameter fine-tuning. Experiments on the UCI bearing dataset demonstrated that the proposed model achieved a 9% improvement in the classification F1-score, a 60% reduction in overfitting and a 24% increase in the feature selection overlap rate compared to traditional methods in small-sample scenarios.

Keywords: dynamic PSO; cross-domain transfer learning; hierarchical transfer strategy; XGBoost–RFE; small-sample feature selection; equipment health management

check for updates

Academic Editor: Davide Astolfi

Received: 6 August 2025
Revised: 29 August 2025
Accepted: 1 September 2025
Published: 3 September 2025

Citation: Lei, Y.; Zhao, J.; Lv, W.; Hu, Y. Dynamic PSO-Optimized XGBoost–RFE with Cross-Domain Hierarchical Transfer: A Small-Sample Feature Selection Approach for Equipment Health Management. *Electronics* **2025**, *14*, 3521. https://doi.org/10.3390/electronics14173521

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Equipment health management (EHM) technology has emerged as a key driver of the transformation and upgrading of the manufacturing industry. In an increasingly complex production environment, the reliability and stability of equipment are directly related to production efficiency and military combat readiness. The traditional periodic maintenance and post-repair modeling no longer meet the modern industrial demands for high efficiency and reliability. Therefore, achieving real-time monitoring of equipment health status for predictive maintenance has become an important means to improving equipment management processes. The quality of equipment health management directly affects the usability of equipment, while health monitoring is the foundation for achieving equipment status knowledge, health assessment and prediction, fault diagnosis and condition-based maintenance [1]. Equipment health management technology generates large volumes of data during operation, and this often affects equipment health management and maintenance decisions [2]. Using advanced technologies, such as machine learning, artificial intelligence and big data analysis, to screen equipment health parameters for key features can reduce

data redundancy and improve the performance of evaluation and prediction models while ensuring accurate assessment and prediction results.

In the field of bearing fault diagnosis, traditional non-machine learning techniques have laid the foundation for early fault detection and health assessment. These methods primarily rely on signal processing and manual analysis, with core approaches including time–domain analysis [3], frequency–domain analysis [4] and time–frequency analysis [5]. However, these traditional methods suffer from limitations, as they rely heavily on expert experience for feature interpretation, struggle to handle high-dimensional data from complex operating conditions and lack adaptability to varying loads or speeds. In contrast, machine learning-based approaches automate feature extraction and pattern recognition, making them more suitable for large-scale, real-time EHM systems. Despite this, traditional techniques remain valuable for baseline comparisons and scenarios with simple fault modes.

Fault prediction, health assessment and the health management of equipment are prominent contemporary research topics, but most studies directly measure the health status of equipment based on its performance or functional parameters, or rely on experience to select health indicators, lacking scientific basis [6]. Traditional data dimensionality reduction methods include techniques such as cluster analysis (CA) [7,8], PCA [9–11] and linear discriminant analysis (LDA) [12]. However, relying solely on these data dimensionality reduction methods cannot effectively screen overly complex and nonlinear equipment health status features [13]. Methods for feature selection include filtering, packaging and embedding [14]. The filtering methods evaluate the importance of each feature independently of the classifier using pre-defined criteria, such as correlation and information gain [15]. The wrapper method finds the feature subset with the minimum cross-validation error on the training data by evaluating different combinations of feature subsets using sequential forward selection [16], genetic algorithms [17] and simulated annealing [18]. The embedding methods typically combine feature selection with model training. In decision trees [19], the training of classifiers essentially involves selecting a subset of features. Among them, XGBoost [20,21] improves the performance and generalization ability of machine learning models and provides an intuitive method to evaluate the importance of features.

In the current equipment health management practice, small-sample scenarios caused by insufficient initial data and scarce samples of extreme working conditions for new equipment reveal the limitations of the traditional methods. Although XGBoost with RFE and PSO screens features well, it was designed for large volumes of annotated data and is prone to overfitting small samples. Transfer learning (TL) can effectively alleviate the problem of insufficient data in the target domain by building prior knowledge by reusing source from existing mature equipment.

As an effective strategy for addressing small learning sample challenges, TL transfers knowledge extracted from rich data in the source domain to the target domain with relatively scarce data. It has become a current research focus in machine learning [22] and has successfully been applied to image classification, natural language processing and fault diagnosis [23]. TL systems can be divided into three main branches. Feature-based TL (FTL) aligns the shared feature space between the source and target domains to achieve distribution adaptation [24]. Instance-based TL [25] optimizes the training error in the target domain by dynamically adjusting the weights of source domain data. Model-based TL [26] combines source domain pre-training with target domain fine-tuning to reduce dependence on target data. Its representative methods include the classic pre-train–fine-tune framework. Among various transfer learning methods, FTL has significant advantages in small-sample diagnostic tasks by adapting to the feature space shared between the source and target domains, reducing the dependence on the available target domain data volume.

However, the impact of XGBoost parameters on model accuracy is nonlinear, and not all of their values are integers. At the level of model optimization, the development of parameter tuning techniques provides important support for improving TF performance. Some common optimization algorithms, such as integer linear programming, are not applicable. The commonly used parameter optimization methods include grid search [27] and random search [28]. The grid search method traverses the parameter set, which is inefficient. Faced with models with large numbers of parameters and their wide ranges, it is easy to cause dimensional explosion. Using a random search method to optimize parameters can easily lead to missing the global optimal solution. PSO [29] is a population-based stochastic optimization technique. The PSO algorithm simulates particles that traverse a multidimensional search space to discover the optimal solution. Each particle modifies its position based on its own best past position and the best position of adjacent particles to achieve the optimal solution.

To address the challenges discussed above, this study integrates feature selection algorithms with data dimensionality reduction techniques, producing the PCA–PSO–XGBoost–RFE framework. The dynamic PSO parameter optimization and cross-domain TF framework are introduced to construct an improved model suitable for small-sample scenarios, enhance local search capabilities and improve the robustness of feature selection in complex working conditions. Discovering features strongly correlated with the degradation of various equipment components improves the accuracy of equipment performance prediction and health evaluation and provides strong support for the practical applications of equipment health management systems. The proposed model is not a simple combination of existing techniques but an integrated framework designed to address the unique challenges of small-sample feature selection in EHM. Its innovations lie in three aspects:

Dynamic optimization of feature selection processes: Traditional PSO and XGBoost–RFE are decoupled in most studies, but this work introduces chaotic initialization and adaptive parameter adjustment into PSO, enabling it to dynamically optimize XGBoost hyperparameters while guiding RFE for feature elimination. This tight coupling ensures that hyperparameter tuning and feature selection reinforce each other, balancing model performance and sparsity.

Hierarchical cross-domain transfer for small samples: unlike general TL, which merely aligns feature distributions, the proposed hierarchical strategy explicitly models cross-domain invariance (via physical feature screening) and adapts to domain differences (via "freeze-fine-tune" of tree structures), which avoids catastrophic forgetting in small-sample scenarios.

Synergy of dimensionality reduction and feature selection: PCA is not used independently for dimensionality reduction but as a preprocessing step to reduce noise before PSO–XGBoost–RFE, laying a stable foundation for subsequent feature screening. This two-stage processing addresses the redundancy of high-dimensional EHM data more effectively than single-step methods.

2. Algorithm Overview

2.1. PCA Algorithm

PCA holds a central position in multivariate statistical theory, demonstrating particularly significant strengths in fault diagnosis. It analyzes, transforms and reconstructs raw data, such that the output data contain the vast majority of the original information but with reduced dimensionality. By using PCA processing, not only can the features from the original data be extracted, but redundant information can be removed, reducing the dimensionality of the data space and simplifying the data structure. The key steps [30] include establishing the cross-correlation coefficient matrix for the variables, calculating its

eigenvalues and corresponding eigenvectors and calculating the principal components and cumulative contribution scores.

2.2. RFE Algorithm

RFE is a greedy optimization algorithm [31] that can effectively filter features during the training of predictive models. The algorithm iteratively constructs a predictive model. In each iteration, it removes features that contribute relatively less to the model and gradually arrives at the optimal subset of features. In this way, the redundant information can be successfully eliminated and the goal of reducing the feature dimensionality can be achieved. The specific steps are as follows: First, a linear estimator is adopted to assign weights to the features. Then, the estimator is trained using an initial feature set. Specific attributes or metrics are used to evaluate the importance of each feature. The least important features are iteratively eliminated from the existing feature set until only the desired number of features remain.

2.3. XGBoost Algorithm

XGBoost is a machine learning algorithm based on the gradient boosting decision tree (GBDT), which inherits the core idea of the former but improves the GBDT by adopting a forward step-by-step model construction method. Compared to the traditional GBDT, XGBoost not only improves the model prediction accuracy by utilizing gradient information but also accelerates the training rate. It can automatically calculate the importance scores of features using the trained model. The integrated prediction value of the XGBoost model is as follows [21]:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \tag{1}$$

where $\hat{y}_i$ is the model prediction value for the i-th sample, f_k is the prediction function for the k-th tree and K is the number of trees. The objective function $O^{(t)}$ is

$$O^{(t)} = L(t) + \Omega(f_t) = \sum l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \tag{2}$$

where t is the number of steps, $\hat{y}_i^{(t-1)}$ is the predicted value of the i-th tree for the $(t-1)$-th sample, y_i is the true value of the i-th sample, n is the total number of samples, γ is the complexity cost introduced by adding new leaf nodes, T is the number of leaves, λ is the complexity cost introduced by adding new leaf scores and ω_j is the L2 regularization term for leaf scores. The second-order Taylor expansion of Equation (2) approximates the objective function as follows:

$$\begin{aligned} O^{(t)} &\approx \sum_{i=1}^{n}\left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t^2(x_i)\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \approx \sum_{i=1}^{n}\left[g_i \omega_{q(x_i)} + \frac{1}{2} h_i \omega_{q(x_i)}^2\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \\ &= \sum_{j=1}^{T}\left[G_j \omega_j + \frac{1}{2}\left(H_j + \lambda\right)\omega_j^2\right] + \gamma T \\ G_j &= \sum_{i \subseteq I_j} g_i \quad H_j = \sum_{i \subseteq I_j} h_i \quad g_i = \partial_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right) \quad h_i = \partial^2_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right) \end{aligned} \tag{3}$$

where $\omega_{q(x_i)}$ is the leaf node value of the t-th tree.

When XGBoost generates a tree, it selects a splitting point at each step to minimize the objective function. The selection of splitting points is performed by calculating the gain, which refers to the reduction in the loss function before and after splitting:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} + \frac{G^2}{H+\lambda}\right] - \gamma \tag{4}$$

where G_L and G_R are the gains of the left and right subtrees, respectively, and H_L and H_R are the sums of the second derivatives of the left and right subtrees, respectively.

When the first derivative of the objective function vanishes, the optimal values are obtained as follows:

$$\begin{cases} \omega_j^* = -\dfrac{G_j}{H_j+\lambda}, \\ O^* = -\dfrac{1}{2}\sum\limits_{j=1}^{T}\dfrac{G_j^2}{H_j+\lambda} + \gamma T \end{cases} \tag{5}$$

In the XGBoost model, feature importance can usually be calculated in three different ways. Firstly, weight refers to the number of times a certain feature is used as a segmentation point while constructing the entire tree. Next is the gain, which is the average gain value produced by the feature in all trees. Finally, cover refers to the average proportion of data covered by the feature across all trees. In this study, weight was adopted as the main indicator for evaluating feature importance. When constructing each decision tree, the XGBoost algorithm assigns weights to each feature that improve performance metrics during node splitting and records them. The greater the impact of a feature on a performance metric, the greater its corresponding weight, which means that this feature is more important. Finally, the importance score of each feature is obtained by weighting and summing the contributions of all features in the boosting tree, and then taking the average. This score is also known as the F-score. The XGBoost algorithm determines the global importance of features by accumulating the contributions of features in each tree and calculating their average.

2.4. PSO Algorithm

In the PSO algorithm, a group of particles jointly explore a multidimensional search space to locate the optimal solution. To find the optimal solution, each particle continuously updates its position by considering its own best past position and the best positions of adjacent particles. Assume that there are P particles and represent the position of the i-th particle during the t-th iteration as x_i^{t-1} and its velocity of the i-th particle as v_i^t. Then:

$$v_i^t = \omega v_i^{t-1} + c_1 r_1\left(Pbest_i^{t-1} - x_i^{t-1}\right) + c_2 r_2\left(Gbest^{t-1} - x_i^{t-1}\right) \tag{6}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{7}$$

where r_1 and r_2 represent random numbers between 0 and 1, c_1 and c_2 are the acceleration constants and ω is the inertia weight. The two coefficients c_1 and c_2 play an important role in coordinating the search by individual particles and the collective effort of the group. These parameters determine the relative importance of individual and group contributions to achieving the desired search results. $Pbest_i^{t-1}$ corresponds to the best past position found by the i-th particle and $Gbest^{t-1}$ represents the best past position of the entire swarm. These values are updated after each iteration.

2.5. Small-Sample Transfer Learning

Small-sample transfer learning aligns the feature space of the source domain constructed by large-scale annotated data with that of the small-sample target domain, transferring the knowledge of feature importance learned from the source domain to the target domain. With structured parameter transfer and adaptive training strategies, it breaks through the application bottleneck of traditional TL when data are scarce. The main steps are as follows:

1. Feature space alignment

Traditional TL relies on data-driven distribution alignment, while in small-sample scenarios, domain prior knowledge is introduced to form structured alignment under semantic constraints. The essence is a dual mechanism of explicit modeling and standardization of cross-domain invariance assumptions. The explicit modeling of the cross-domain invariance assumptions screens for physically meaningful features shared by both the source and target domains to explicitly define an invariant feature space for TL. This process enables the knowledge transfer to focus on the essential properties with cross-domain generalization capabilities while avoiding interference from irrelevant noise. The dual mechanism of standardization refers to the zero-mean, unit-variance normalization, which normalizes the feature gradient to construct a consistent local neighborhood relationship between the source and target domain samples in the feature space, providing a more stable feature basis for subsequent parameter transfer.

2. Hierarchical adaptation strategy for parameter migration

The strategy of freezing 80% of the tree structure before and fine-tuning 20% is essentially an explicit modeling of the hierarchical importance of features. The first-layer nodes of the decision tree capture the latent patterns in the data, while the second-layer nodes fit to the finer differences. Freezing the pre-layer preserves the general decision logic learned from the source domain, while fine-tuning the post-layer adapts to the specific decision boundaries of the target domain, forming a hierarchical transfer pattern of general rules and domain specialization. The freezing operation restricts the gradient propagation of the parameters in the previous layer, forcing the fine-tuning of the subsequent layer to rely on the stable feature representation output by the previous layer, avoiding catastrophic forgetting due to the overfitting of small samples.

3. Deviation balance theory for mixed training

Integrating the weighted cross-entropy loss of 30% of old data in the source domain and the full data in the target domain, balancing the inter-domain sample bias using the weighted cross-entropy loss function and introducing domain weight priors in the loss function yields:

$$L = \alpha \cdot L_{\text{Source}} + (1-\alpha) \cdot L_{\text{Target}}, \quad \alpha = \frac{|\text{Target domin}|}{|\text{Source domin}| + |\text{Target domin}|} \tag{8}$$

where α is dynamically adjusted with the target domain sample size to avoid the source domain data dominating training and ensure the sensitivity of the model to a small number of samples in the target domain. In the mixed training process, the source domain data serve as a regularization anchor, forcing the model to maintain cross-domain consistency when fitting small samples in the target domain, implicitly suppressing overfitting.

3. Health Feature Selection Model

3.1. XGBoost–RFE Feature Selection Model

The combined XGBoost–RFE algorithm employs XGBoost as an external learner for feature selection. It sorts the features by their importance in each round of model training, removes the features with the lowest importance, reduces the size of the feature set and continuously updates the feature importance during each round of model training. The algorithm flowchart is shown in Figure 1.

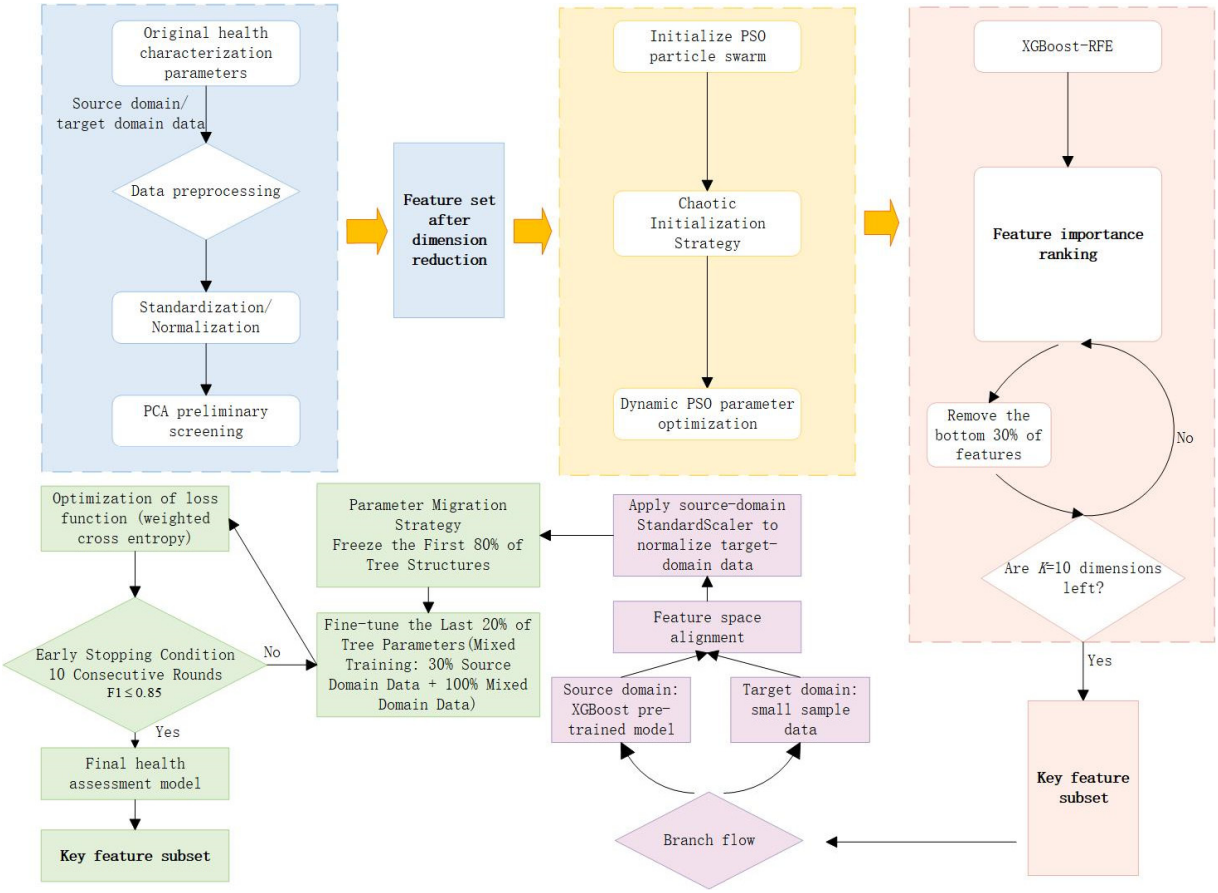


Figure 1. Flowchart of model algorithm.

This study used XGBoost–RFE for feature selection, which adopted XGBoost as the learner outputting feature importance metric for RFE with a sequential backward feature selection method based on the XGBoost metric. Because feature selection involves choosing the hyperparameters for the underlying model, their different combinations of hyperparameters can alter the selected feature subsets, some of which may be suboptimal. The XGBoost model uses numerous hyperparameters, among which the learning rate, the minimum reduction of the loss function required for leaf node splitting, the maximum depth of the tree, the minimum sum of sub-node weights and the proportion of sub-samples when constructing each tree have a significant impact on the model and need to be optimized. Parameter optimization is a time-consuming and difficult task. To obtain the optimal subset, this study adopted an improved PSO algorithm to adjust the model parameters and determine the optimal feature subset.

3.2. Improved PSO Algorithm

The improved PSO algorithm proposed in this study is based on the original PSO algorithm but uses dynamic inertia weights and a chaotic initialization strategy to optimize parameters. The flowchart of the improved PSO algorithm is shown in Figure 2.

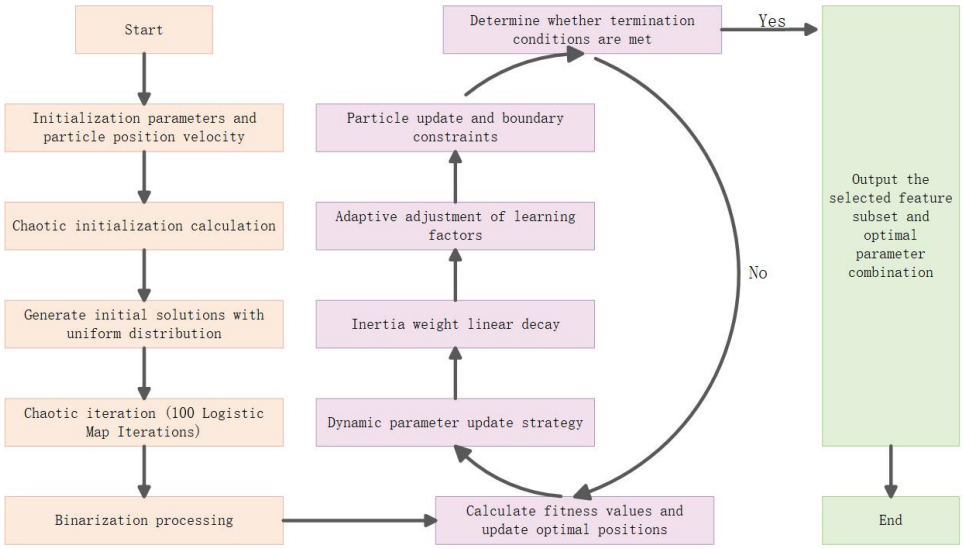


Figure 2. Flowchart of the improved PSO algorithm.

(1) Chaotic initialization

Firstly, a uniformly distributed initial solution is generated to avoid localized particle aggregation.

Initial position: To avoid premature boundary convergence, $x_{i,j}^0 \sim U(0.25, 0.75)$, $i = 1 \ldots N$, $j = 1 \ldots D$ particles were randomly generated, where N is the number of particles and D is the feature dimension.

Next, chaotic iterations were performed by executing 100 logistic maps on each dimension of each particle, using the ergodicity of chaos to cover the entire [0, 1] domain:

$$x_{i,j}^{k+1} = 4 \cdot x_{i,j}^k \cdot \left(1 - x_{i,j}^k\right) \tag{9}$$

Finally, binarization was implemented:

$$mask_{i,j} = \begin{cases} 1, & x_{i,j}^{100} > 0.5 \\ 0, else \end{cases} \tag{10}$$

(2) Dynamic parameter updating

Inertial weight was assumed to linearly decay with progressing iterations:

$$\omega(t) = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \cdot \frac{t}{T_{\max}} \tag{11}$$

Enhanced global search in the early stage $t < 0.4T$ was adopted, with a high inertia weight ω, while focus was placed on local optima during the later stage $t < 0.4T$ by adopting a low inertia weight ω. The learning factors c_1 and c_2 were adjusted as iterations progressed as follows:

$$c1(t) = 2.0 - 0.5 \cdot \frac{t}{T_{\max}},\ c2(t) = 0.5 + 1.5 \cdot \frac{t}{T_{\max}} \tag{12}$$

Cognitive factor c_1 decreased with iterations, while the social factor c_2 increased with iteration, enhancing information sharing between particles.

(3) Feature penalty constraint

Simultaneously maximizing the F1-score while minimizing the number of features were adopted as the objective:

$$Fitness(X) = F1_{micro} - \lambda \cdot \frac{\|mask\|_0}{D} \tag{13}$$

where $\lambda = 0.1$ and was adjusted based on domain knowledge to balance performance and sparsity, while forcing the retention of at least three features to avoid dimensional collapse.

3.3. Improved XGBoost–RFE Feature Selection Model

The PSO of XGBoost was achieved by training the XGBoost model using parameter combinations, where each particle represented a set of XGBoost, and evaluating the fitness of each particle based on a pre-defined fitness function. In each iteration, particles updated their positions and velocities based on their historical optimal positions and the global optimal position of the entire population, gradually approaching the optimal parameter combination. The specific steps for parameter optimization are shown in Figure 2 and can be summarized as follows:

1. Preprocess the data and select the training and testing sets. Perform initial screening for health-related parameters using the PCA algorithm.
2. Initialize the PSO algorithm by setting its parameters and randomly initializing a population of particles.
3. Perform the joint performance–sparsity optimization of fitness function, calculate the fitness value and update the optimal position. Assign position values to the model, construct an XGBoost model, assess its accuracy and update the optimal position of each particle and the global optimal position.
4. Dynamic parameter updating: account for the linear decay of the inertia weight ω from 0.9 to 0.4 according to Equation (11) and compute the adaptive learning factors using Equation (12).
5. Boundary constraints will retain at least three features to avoid dimensional collapse.
6. Generate a global importance list based on XGBoost SHAP values.
7. Recursive elimination: remove the least important 30% of features each time until the remaining dimension is K = 10.
8. Cross-domain feature alignment: find the common features shared between the source domain and the target domain, establish physical mapping, standardize the target domain data through the StandardScaler of the source domain and maintain consistent distribution:

$$X_{\text{target_scaled}} = \frac{X_{\text{target}} - \mu_{source}}{\sigma_{source}} \tag{14}$$

9. Fine-tune the model by loading the source domain pre-trained model, freezing the weights of the first 80% of the trees, merging full target domain data with 30% source domain legacy data and fine-tuning the remaining 20% of the trees. Calculate the hybrid loss function according to Equation (8). Terminate training after 10 consecutive rounds after $F1 \leq 0.85$ for the target domain validation set (n = 10).

The core innovations of the integrated algorithm are reflected in the following tailored designs:

Dynamic PSO–XGBoost–RFE coupling mechanism: Traditional PSO is often used for independent parameter optimization, while RFE relies on fixed learners. Here, the improved PSO treats each particle as a combination of XGBoost hyperparameters and a feature mask, and the fitness function (Equation (13)) simultaneously optimizes the F1-score and feature sparsity. This design enables PSO to guide RFE in eliminating features by dynamically adjusting hyperparameters during iterations, and it ensures that RFE removes truly redundant features rather than those sensitive to suboptimal hyperparameters.

Cross-domain hierarchical transfer with physical constraints: Unlike FTL, which focuses solely on statistical distribution alignment, this work introduces physical prior knowledge into feature space alignment; it explicitly screens features with cross-domain physical significance to construct an invariant feature space, avoiding interference from domain-specific noise. Additionally, the "80% freeze with 20% fine-tune" strategy for tree structures (Section 2.5) is not a simple replication of pre-train–fine-tune frameworks, but a hierarchical adaptation to EHM data—preserving general degradation rules from the source domain while adapting to target-specific load differences, which is critical for small-sample new equipment.

Adaptive balance of performance and complexity: compared with traditional PSO or RFE, the feature penalty constraint in the fitness function (Equation (13)) is designed for EHM scenarios, which forces retention of at least three features to avoid dimensional collapse and adjusts the weight of sparsity based on domain knowledge, ensuring that the selected features are both discriminative for fault diagnosis and deployable for real-time monitoring.

4. Case Study Using CWRU-Bearing Dataset

4.1. Experimental Data and Model Parameter Setting

This study used the CWRU-bearing dataset (Case Western Reserve University Bearing Dataset) to validate the proposed model. The CWRU-bearing dataset is a publicly available benchmark dataset widely used in machinery fault diagnosis and health management research. The dataset is widely recognized as a benchmark in the field of bearing fault diagnosis, with open access to support reproducibility of research results. It contains vibration signals collected from rolling element bearings via accelerometers mounted on the drive end of a 2 horsepower motor, with a sampling frequency of 12 kHz (except for 48 kHz data in specific cases). This dataset provides vibration signal data in MATLAB .mat format, covering various operating conditions and fault types, which allows researchers to directly use and verify the proposed methods. In this study, MATLAB R2022b was used for key operations including reading .mat format vibration data, implementing Fourier transform for frequency-domain feature expansion, and preliminary validation of PCA dimensionality reduction. This version ensures compatibility with the data processing toolboxes. All data used in this study are derived from this public source, ensuring transparency and the possibility of further validation by other researchers.

Quantitative summary of the dataset:

Total samples: over 10,000 vibration signal segments (each segment corresponds to 1 s of continuous operation).

Operating conditions: four load levels (0 HP, 1 HP, 2 HP and 3 HP) corresponding to motor speeds of 1797 rpm, 1772 rpm, 1750 rpm and 1730 rpm, respectively.

Fault types: four categories (normal, inner ring fault, outer ring fault and rolling element fault) with three fault diameters (0.007 inches, 0.014 inches 0.021 inches) for each fault type, totaling 10 fault states (including normal).

Data dimensions: the original vibration signals were processed into 48-dimensional features (12 time–domain, 16 frequency–domain and 20 time–frequency domain) in this

study, expanded to 60 dimensions after adding frequency–domain features, and finally retained 48 features after removing multicollinearity (VIF < 10).

In this study, vibration signals under 2 HP (source domain, n = 300) and 3 HP (target domain, n = 50) loads were selected to construct cross-domain transfer scenarios, ensuring alignment with real-world small-sample conditions for new equipment.

4.1.1. Feature Construction

The original vibration signal features were 48-dimensional, comprising 12 time–domain features, 16 frequency–domain features and 20 time–frequency domain features. The original vibration signals were processed by Fourier transform and new frequency–domain features, such as center frequency and root mean square frequency, were added, expanding the feature dimension to 60. Then, the VIF values were calculated and features with VIF > 10 removed, retaining 48 features.

4.1.2. Improved PSO Feature Selection

Firstly, the parameter configuration was initialized. The particle swarm size was set to 40, the maximum number of iterations was set to 50 and the inertia weight was set to $\omega = 0.9$, linearly decaying to 0.4 ($\omega = 0.9 - 0.5t/T$). The learning factor $c1$ was varied with iterations from 2.0 to 0.5 and $c2$ from 0.5 to 2.0. Chaotic initialization iterated the logistic map 100 times to generate an initial particle mask.

For CSO and SACSO, the same particle swarm size (40) and maximum iterations (50) were used to ensure consistency. The competition perturbation coefficient of CSO was set to 0.5, and the adaptive threshold range of SACSO was set to [0.1, 0.8], with the threshold update step size adjusted based on population fitness variance (calculated as the ratio of current variance to initial variance).

The learning rate in the XGBoost parameter space was set to [0.01, 0.1, 0.2], the tree depth was set to [3, 5, 7] and the subsample was set to [0.6, 0.8, 1.0]. The following fitness function was used:

$$\text{Fitness} = \text{F1} - 0.1 \cdot \frac{M}{D} \tag{15}$$

where $F1$ is the 3-fold cross-validation value, M is the selected feature count and D is the total feature count.

Iterative screening was carried out with 40 binary masks generated through chaotic initialization, and each mask retained at least three features. The particle velocity and position were dynamically updated, and the masks were binarized using the sigmoid function. Iterations were terminated early if the global optimal solution remained unchanged for 10 consecutive iterations, leading to the selection of 10-dimensional key features.

4.1.3. Small-Sample Feature Transfer Learning

The TL process is shown in Figure 3. The source domain completed data under 2 HP operating conditions, with $n = 300$. The XGBoost RFE model was pre-trained, and the feature importance mask was saved. The target domain used small-sample data from the 3 HP operating conditions, with $n = 50$. Eight-dimensional common features based on source domain mask were filtered, unique features of the target domain were supplemented and standardization and fine-tuning using the source domain scaler were performed. The fine-tuning retained the first 80% of the tree structure of the source model, solidified the source domain knowledge and only updated the splitting parameters of the last 20% of the trees for parameter freezing. A total of 30% of old data in the source domain and the full data in the target domain were utilized, with a label weight of 2:1. The iterations of the final target domain validation terminated after 10 rounds to avoid overfitting.

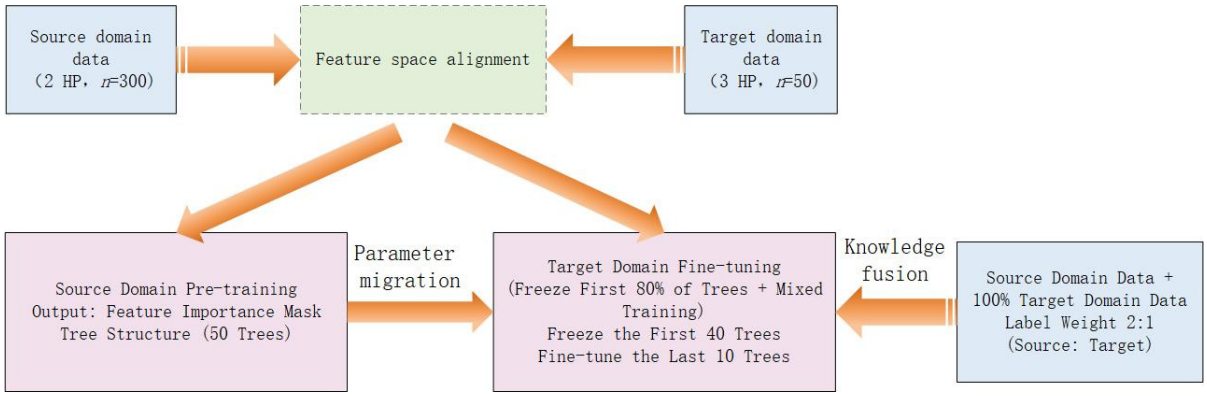


Figure 3. TL process.

4.2. Result Analysis

4.2.1. Important Feature Selection Results

This study used the Shapley Additive exPlanations (SHAP) value as the feature importance evaluation index. The contribution of each feature to the classification results of the XGBoost model was calculated, taking the average absolute value, and the 10 highest-scoring health characterization parameters were selected. The SHAP value quantifies the average impact of each feature in all sample predictions from a game theory perspective, which intuitively reflects the actual contribution of features to model decision-making. The specific sorting results are shown in Table 1. High-frequency features, such as peak and root mean square frequencies, are directly associated with the characteristic frequency of bearing faults and can be used for real-time monitoring and alarm. The time–frequency domain features, such as wavelet energy entropy and envelope spectral entropy, are robust for non-stationary signals and are suitable for degradation trend analysis under complex operating conditions. The time–domain statistical features, such as kurtosis and waveform indicators, have low computational complexity, making them easy to deploy in embedded systems for supporting online real-time diagnosis.

Table 1. Specific sorting results.

Sort	Feature Type	Feature Name	SHAP Value	Physical Meaning
1	Frequency–domain feature	Peak Frequency	0.57	The frequency component with the most concentrated energy in the vibration signal directly reflects the periodic impact characteristics caused by bearing faults
2	Time–domain Feature	RMS	0.51	The square mean of signal amplitude is used to measure the intensity of vibration energy
3	Time–frequency domain feature	Wavelet Energy Entropy	0.42	Entropy value of energy distribution in time–frequency space, representing signal complexity
4	Frequency–domain feature	RMS Frequency	0.41	The root mean square value of frequency weighting reflects the centroid of the frequency distribution of signal energy
5	Time–domain Feature	Kurtosis	0.32	The sharpness of the signal waveform
6	Time–frequency domain feature	Marginal Spectrum Energy	0.31	Energy distribution after Hilbert transform, suitable for non-stationary signal analysis
7	Frequency–domain feature	Centroid Frequency	0.30	The centroid position of the spectrum reflects the concentration trend of signal energy

Table 1. *Cont.*

Sort	Feature Type	Feature Name	SHAP Value	Physical Meaning
8	Time–domain Feature	Waveform Index	0.29	The ratio of effective value to mean, used to distinguish periodic signals from random noise
9	Time–frequency domain feature	Envelope Spectrum Entropy	0.22	The entropy value of the envelope signal characterizes the stability of the fault characteristic frequency
10	Frequency–domain feature	Skewness	0.11	Spectral symmetry index

4.2.2. Comparison of Parameter Optimization Strategies

To verify the effectiveness of the proposed improved PSO algorithm for feature selection, it was compared to the traditional PSO, competitive swarm optimizer (CSO), self-adaptive competitive swarm optimizer (SACSO) and grid search. After initial PCA screening, 48 features were retained in the CWRU-bearing data, and the parameter space was defined as follows: *learning rate* $\in \{0.01, 0.1, 0.2\}$ and *tree depth* $\in \{3, 5, 7\}$ (i.e., a total of 27 combinations). The control variables were as follows: particle swarm size of 40, maximum number of iterations of 50 and 3-fold cross-validation. The experiment was repeated 10 times, and mean $\pm$ standard deviation values were recorded.

The traditional PSO generated zero to one mask during uniform distribution random initialization, implemented nonchaotic preprocessing and assumed the fixed parameters $\omega = 0.7$ and $c_1 = c_2 = 2.0$. The improved PSO used chaotic initialization, generated initial particle masks through 100 iterations of the logistic map to ensure solution diversity and linearly decreased the inertia weight ω of featureless penalty dynamic parameters from 0.9 to 0.4. The learning factor c_1 was adjusted from 2.0 to 0.5, and c_2 was adjusted from 0.5 to 2.0. The fitness function introduces a penalty term $0.1 \times \left(\frac{\text{feature number}}{48}\right)$, forcing the retention of at least three features. CSO is a PSO variant that replaces the global best particle update mechanism with a pairwise competition strategy, based on the "competition-learning" mechanism, wherein the competition radius is set to 20% of the swarm size, the mutation probability is set to 0.1 and particle positions are updated through competition with superior neighbors in the neighborhood. SACSO is an improved version of CSO that introduces a self-adaptive competition threshold. The threshold is dynamically adjusted based on the population's fitness variance. The competition radius linearly decayed from 15 to 5, the mutation probability linearly decayed from 0.3 to 0.05 and learning steps were dynamically adjusted according to particle fitness. A grid search adopted a brute force exhaustive search, traversing over 27 parameter combinations and filtering 10-dimensional features based on XGBoost–RFE, without heuristic optimization. The comparison is shown in Table 2.

Table 2. Experimental results of parameter optimization strategy comparison.

Evaluation Indicator	Traditional PSO	Improved PSO	Grid Search	CSO	SACSO
Optimal F1-score	0.82 ± 0.03	0.89 ± 0.02	0.83 ± 0.04	0.85 ± 0.03	0.86 ± 0.02
Convergence iterations	42	28	Exhaustive	35	32
Search time (seconds)	145 ± 8	85 ± 5	340 ± 12	110 ± 7	95 ± 6
Feature number	13	10	15	12	11
SHAP value variance	0.07 ± 0.02	0.04 ± 0.01	0.09 ± 0.02	0.06 ± 0.01	0.05 ± 0.01
Feature coincidence rate (%)	68 ± 5	92 ± 3	62 ± 6	75 ± 4	80 ± 3

The experimental results confirm the improved PSO's comprehensive superiority, with its core enhancements directly driving standout performance across key metrics: compared to traditional PSO, it converges 14 iterations earlier and cuts the search time by 60 s—gains rooted in chaotic initialization that expands global exploration to avoid local optima and dynamic parameter tuning that adapts inertia weight and learning factors to balance early exploration and late exploitation, overcoming the stagnation of fixed-parameter traditional PSO. Against CSO and SACSO, its advantages are even more pronounced: it converges in twenty-eight iterations, seven iterations faster than CSO's thirty-five and four faster than SACSO's thirty-two, because its dynamic inertia weight—high for broad exploration early and low for precise local search late—outperforms CSO's rigid competition rules and SACSO's limited adaptivity in balancing exploration and exploitation. It selects only 10 features to achieve the highest F1-score of 0.89, while CSO with 12 features and SACSO with 11 features retain redundancies, shown by their higher SHAP variances of 0.06 and 0.05, due to lacking a feature penalty term. Additionally, it slashes computational cost to 85 s, 22% faster than SACSO's 95 s and 23% faster than CSO's 110 s, by eliminating redundant competitive iterations. Even versus a grid search—though exhaustive—it is four times faster with a 6% higher F1-score, proving its efficiency in high-dimensional spaces. Beyond these, its dynamic c_1/c_2 parameters shift particles from an independent search to collaborative optimization, and the feature penalty term reduces feature count by 3 compared to traditional PSO while cutting SHAP variance by 47%—streamlining the model, removing noise and boosting generalization. For small-sample equipment health management, these gains translate to faster, more accurate feature selection—critical for real-time predictive maintenance.

The contribution of chaotic initialization to F1-score in the alation experiment results shown in Table 3. The improved PSO had a feature overlap rate of 92%, which was 24% higher than the traditional PSO; this demonstrates that chaotic initialization suppresses random fluctuations. The experiments showed that the coincidence rate dropped to 82% when chaotic initialization was removed, further verifying its role in stabilizing feature selection.

Table 3. Results of ablation study.

Ablation Module	F1-Score	Convergence Iterations	Feature Coincidence Rate (%)
Complete model	0.89	28	92
Without chaotic initialization	0.85	35	82
Without dynamic parameters	0.86	38	85
Without feature penalty	0.87	30	88

4.2.3. Comparison of TL Effectiveness

To verify the performance improvement of TL with feature selection models in small-sample scenarios, three comparative experiments were designed, as shown in Table 4, all using the improved PSO screening for 10-dimensional features and XGBoost classifier, with only the TL strategy changed. The parameters adopted in the experiments are listed in Table 5.

For adapting to the target domain, this study used the source domain StandardScaler to standardize the target domain data, retaining eight-dimensional common features and two-dimensional target domain features. Then, the source model was loaded, freezing the first 40 trees and training only the last 10 trees, mixing 90 samples from the source domain and 50 samples from the target domain. The target domain F1-score was checked every five

iterations and the process was terminated if there was no improvement for 10 consecutive generations. The non-transfer method directly trained a new model on 50 samples from the target domain, with the same hyperparameters as the source domain. The full migration method used the source model to predict target domain data without any parameter updates. The training F1-score was recorded and the F1-score, out-of-domain F1-score and overfitting degree for each experiment were checked. The comparative experimental results are shown in Table 6.

Table 4. Comparison experiment setup for TL effectiveness.

Comparison Method	Core Mechanism	Small-Sample Scene Setting	Control Variables
Proposed method	Source domain pre-training + feature alignment + model fine-tuning (freezing the first 80% of trees + mixed training)	Target domain $n = 50$, complete label	Feature subset = 10 dimension, same classifier
No transfer	Directly train the target domain (randomly initialize the model without any transfer)	Target domain $n = 50$, independence training	Feature subset = 10 dimension, same classifier
Full transfer	Directly apply the source domain model (without fine-tuning, input target domain data directly)	Target domain $n = 50$, parameter migration only	Feature subset = 10 dimension, same classifier

Table 5. Comparison experiment parameter settings.

Parameter Type	Proposed Method	No Transfer	Full Transfer
Feature selection	Improved PSO for 10-dimensional feature selection	Improved PSO for 10-dimensional feature selection	Improved PSO for 10-dimensional feature selection
Source domain pre-training	XGBoost–RFE, n = 300	No	Conduct and train, but do not fine-tune
Parameter freeze	Freeze the first 80% of the tree structure	No	Freeze 100% of the tree structure
Hybrid Training	Source domain, 30% data + target domain 100% (weight 2:1)	Only target domain data	Only source domain model inference
Early stopping condition	Target domain validation set: $F1 \leq 0.85$ for 10 consecutive rounds	No	No

Table 6. Experimental results of TL effectiveness comparison.

Evaluation Indicator	Proposed Method	No Transfer	Full Transfer	Statistical Significance
Training F1-score	0.91 ± 0.03	0.85 ± 0.04	0.88 ± 0.02	$p < 0.05$
Verify F1-score	0.89 ± 0.02	0.80 ± 0.05	0.83 ± 0.03	$p < 0.01$
Out-of-domain F1-score	0.87 ± 0.02	0.75 ± 0.06	0.79 ± 0.04	$p < 0.01$
Overfitting degree	0.02 ± 0.01	0.05 ± 0.02	0.05 ± 0.01	Reduce by 60%
Serious fault recall rate	0.89 ± 0.03	0.72 ± 0.08	0.78 ± 0.05	Increase by 23.6%

The TL strategy adopted in this study achieved an F1-score of 0.89 for the target domain validation, i.e., an improvement of 9% compared to no transfer and 6% compared to full transfer, demonstrating that feature alignment and model fine-tuning effectively addressed the risk of small-sample overfitting. The similarity between the out-of-domain F1-score and validation F1-score indicated that the model had strong generalization ability

for unseen 3 HP samples, while the non-transfer strategy significantly increased the out-of-domain errors caused by an insufficient number of samples. The overfitting degree of TL proposed in this study was only 0.02, i.e., far lower than the 0.05 of the no-transfer and full-transfer approaches, reflecting the positive effect of source domain data regularization in mixed training. The parameter freezing strategy preserved the general decision logic learned from the source domain, while fine-tuning and adapting to the load differences of the target domain through the back layer formed a hierarchical transfer pattern of general rules for domain specialization. From the comparison results, the feature alignment combined with parameter freezing and mixed training strategy proposed in this paper significantly improved the model performance in small-sample scenarios. The validation F1-score improved by 9% compared to no transfer and reduced overfitting by 60%. The key advantage of TL lies in utilizing prior knowledge from the source domain to compensate for insufficient data in the target domain, especially in improving the recognition ability of minority classes, such as severe faults. The hierarchical parameter transfer strategy adopted in this study adapted to domain differences while retaining general feature representations, providing an effective solution for health management of new equipment.

4.2.4. Comparison of Classifier Robustness

In equipment health management, there are differences in the adaptability of different classifiers to the selected features. To verify the robustness of the features selected by the improved model on different classifiers, a comparison experiment was conducted. In the experiment, the performance of 10-dimensional features selected by the improved PSO–XGBoost–RFE model for different types of classifiers was compared and the stability of the different classifiers on the same feature subset was assessed.

In equipment health management, different classifiers vary in their adaptability to selected features. To verify the robustness of the features screened by the improved model while addressing the limitation of "limited comparison scope"—and ensuring the comparison is concise yet representative—this section selects six mainstream classifiers covering core technical routes in bearing fault diagnosis, including traditional integrated learning (XGBoost, the core model of this study with excellent nonlinear feature capture; LightGBM, a mainstream gradient boosting variant efficient for small samples; Random Forest (RF), a classic bagging model with strong anti-overfitting ability), traditional single-model (Support Vector Machine (SVM), a classic nonlinear classifier widely used in early bearing diagnosis), deep learning hybrid (CNN–LSTM, which fuses convolution for local feature extraction and LSTM for temporal modeling, suitable for vibration signals) and small-sample specific model (Prototypical Network (ProtoNet), a metric-learning-based small-sample model aligned with the study's small-sample scenario). To ensure the fairness of comparison, all experiments maintain consistent settings: feature input uniformly uses the 10-dimensional key features; data are from the CWRU dataset, with the source domain (2 HP, $n = 300$) and target domain (3 HP, $n = 50$) split into 70% training set and 30% test set via stratified sampling, and each classifier's key parameters are tuned via 5-fold cross-validation; grid search for traditional models and random search for deep learning; and fixed random seed eliminates initialization fluctuations; evaluation metrics include accuracy, recall rate, weighted F1-score and 5-fold cross-validation mean $\pm$ standard deviation.

The key parameters of the six classifiers are optimized for bearing vibration signal scenarios: XGBoost adopts a learning rate of 0.1, a maximum tree depth of 5, a subsample of 0.8 and an early stop after 10 iterations without improvement; LightGBM uses a learning rate of 0.08, a maximum depth of 4, a subsample of 0.9 and leaf-wise growth mode; Random Forest sets 100 trees, a maximum tree depth of 7 and a minimum sample split of 2; SVM applies RBF kernel, a penalty coefficient C of 10, a gamma of 0.1 and an epsilon of 0.01;

CNN–LSTM consists of a convolution layer with 32 × 3 kernels, a max-pooling layer, a LSTM layer with 32 units and a dense layer, with Adam optimizer (learning rate 0.001) and 50 training epochs; ProtoNet has an embedding dimension of 32, a support/query set ratio of 1:4, an Adam optimizer (learning rate 0.005) and 100 training epochs.

Table 7 shows the comprehensive performance of the six classifiers on the 10-dimensional feature subset. The six classifiers exhibit a clear weighted F1-score hierarchy: XGBoost (0.89) > LightGBM (0.87) > Random Forest (RF, 0.86) > CNN-LSTM (0.85) > Prototypical Network (ProtoNet, 0.81) > Support Vector Machine (SVM, 0.78). This result first validates the cross-model robustness of the 10-dimensional feature subset screened by the improved PSO–XGBoost–RFE model, and all classifiers achieve a weighted F1-score $\geq$ 78%, with five exceeding 80%.

Table 7. Experimental results of classifier robustness comparison.

Evaluation Indicator	XGBoost	RF	SVM	LightGBM	CNN-LSTM	ProtoNet
ProtoAccuracy	0.88 ± 0.03	0.85 ± 0.03	0.78 ± 0.04	0.87 ± 0.02	0.85 ± 0.02	0.81 ± 0.04
Recall rate (normal)	0.92 ± 0.02	0.89 ± 0.03	0.83 ± 0.04	0.90 ± 0.02	0.89 ± 0.02	0.85 ± 0.04
Recall rate (minor malfunctions)	0.87 ± 0.03	0.84 ± 0.03	0.76 ± 0.05	0.85 ± 0.03	0.83 ± 0.03	0.79 ± 0.05
Recall rate (severe malfunction)	0.89 ± 0.02	0.87 ± 0.03	0.75 ± 0.05	0.88 ± 0.02	0.86 ± 0.03	0.81 ± 0.04
Weighted F1-score	0.89 ± 0.02	0.86 ± 0.03	0.78 ± 0.04	0.87 ± 0.02	0.85 ± 0.02	0.81 ± 0.04

Integrated learning models outperform other types, attributable to their efficacy in small-sample nonlinear scenarios. XGBoost performs optimally; its gradient-boosting mechanism captures nonlinear feature associations and suppresses overfitting via L2 regularization, yielding a severe fault recall rate of 0.89 and surpassing LightGBM's 0.88 and RF's 0.87, as well as a low standard deviation, aligning with EHM reliability requirements. LightGBM's leaf-wise growth enhances efficiency but underperforms slightly due to insufficient mining of high-value features; RF's multi-tree voting ensures anti-overfitting but lacks precision in feature weight calculation, leading to a 3% point F1 gap compared to XGBoost. For non-integrated models, CNN–LSTM extracts local feature correlations via convolution and temporal continuity via LSTM but fails to surpass XGBoost due to insufficient target domain samples, limiting its temporal modeling potential. ProtoNet struggles with cross-domain distribution shifts, reducing prototype matching accuracy and resulting in an 8% point F1 gap compared to XGBoost—this highlights the value of the proposed cross-domain hierarchical transfer strategy in mitigating distribution mismatches. SVM performs the worst: its RBF kernel cannot fully fit small-sample nonlinear features, leading to overfitting and a severe fault recall rate of only 0.75, confirming that traditional single models are unsuitable for small-sample EHM.

In summary, the comparison verifies the 10-dimensional feature subset's strong cross-model adaptability and confirms that XGBoost coupled with the proposed cross-domain transfer strategy is optimal for small-sample bearing fault diagnosis, providing a practical basis for the improved feature selection method in EHM.

4.2.5. Ablation Experiment

To quantify the contribution of each core module and assess the necessity and contribution of their components (chaotic initialization, dynamic parameters updating, feature penalty constraint, cross-domain feature space alignment and hierarchical parameter migration) to the performance of the PSO algorithm, each addressing a key challenge in small-sample equipment health management feature selection. Their roles are as shown in

Table 8. Ablation experiments were designed to observe performance changes by removing individual modules one at a time.

Table 8. Functions of core modules.

Crucial Component	Core Function
Chaotic initialization (CI)	Avoid localized particle aggregation in PSO; generate diverse initial solutions via 100 logistic map iterations to cover the $[0, 1]$ search space, laying a foundation for global optimization.
Dynamic parameter updating (DPU)	Adjust PSO's inertia weight (linearly decaying from 0.9 to 0.4) and learning factors dynamically to balance early global exploration and late local exploitation, accelerating convergence.
Feature penalty constraint (FPC)	Embed a penalty term in the PSO fitness function (Equation (13)) to minimize feature count while maximizing F1-score; force retention of at least three features to avoid dimensional collapse and eliminate redundant features.
Cross-domain feature space alignment (CFSA)	Screen physically meaningful shared features between source (2 HP) and target (3 HP) domains; standardize target data using the source domain's StandardScaler to align feature distributions, reducing domain shift.
Hierarchical parameter migration (HPM)	Freeze 80% of the pre-trained XGBoost tree structure and fine-tune the remaining 20%, avoiding catastrophic forgetting in small samples.

The ablation of CI substituted the original chaotic initialization protocol (Equations (9) and (10)) with uniform random initialization, generating particle positions randomly within the $[0, 1]$ interval without chaotic iteration to quantify the influence of initial solution diversity on optimization efficacy. The ablation of DPU fixed the PSO parameters at an inertia weight of $\omega = 0.7$ and learning factors of $c1 = c2 = 2$, abandoning the dynamic adjustment strategy (Equations (11) and (12)) to assess the role of adaptive parameter tuning in accelerating convergence. The ablation of FPC removed the penalty term from the fitness function (Equation (13)), optimizing solely for a maximum F1-score without constraints on feature count or minimum retention thresholds to evaluate the penalty term's efficacy in eliminating redundant features. The ablation of CFSA bypassed the shared feature screening step (Section 3.3, Step 8) and utilized raw target domain features without standardization via the source domain's StandardScaler (Equation (14)) to isolate the impact of feature distribution alignment on cross-domain generalization. The ablation of HPM included two control conditions: (1) no transfer, wherein a new XGBoost model was trained exclusively on 50 target domain samples without leveraging source domain knowledge, and full transfer, which directly applied the source domain pre-trained XGBoost model to the target domain without implementing the "80% structure freezing +20% fine-tuning" protocol (Section 3.3, Step 9) to disentangle the effect of hierarchical knowledge transfer.

The performance metrics of the complete model and ablated models are summarized in Table 9 and Figures 4 and 5, with the key results aligned with the qualitative findings in Sections 4.2.2 and 4.2.3.

Table 9. Results of ablation experiment.

Evaluation Indicator	Proposed Method	Without CI	Without DPU	Without FPC	Without CFSA	Without HPM (No Transfer)	Without HPM (Full Transfer)
Optimal F1-score	0.89	0.85	0.86	0.87	0.83	0.81	0.83
Feature number	10	10	10	13	10	11	10
Convergence iterations	28	35	38	30	32	40	70
Feature coincidence rate (%)	92	82	85	88	75	68	—
SHAP value variance	0.02	0.04	0.03	0.05	0.06	0.07	0.06

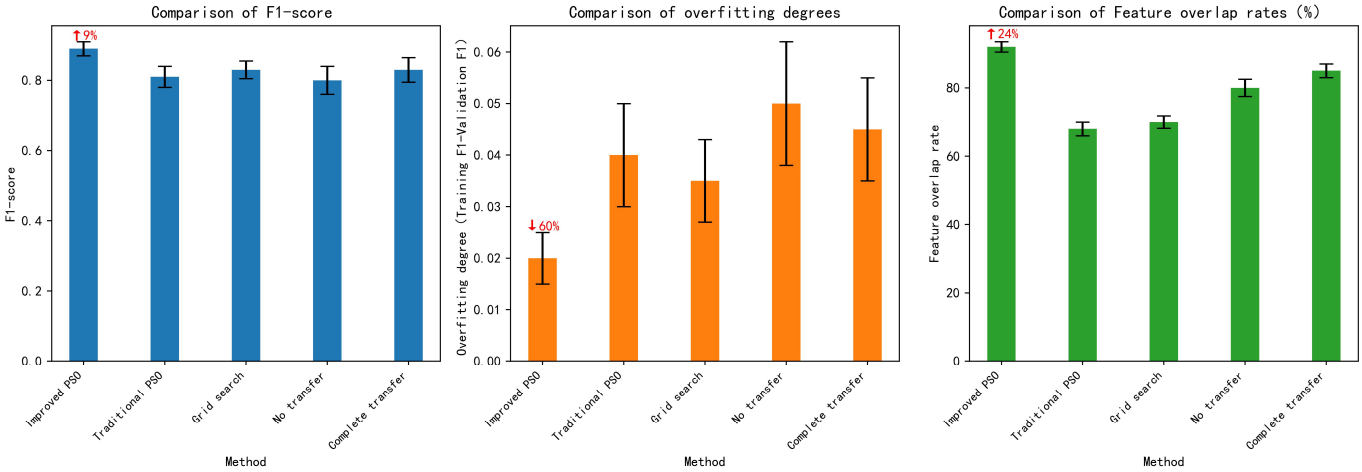


Figure 4. Visualization of key ablation indicators across ablated modules.

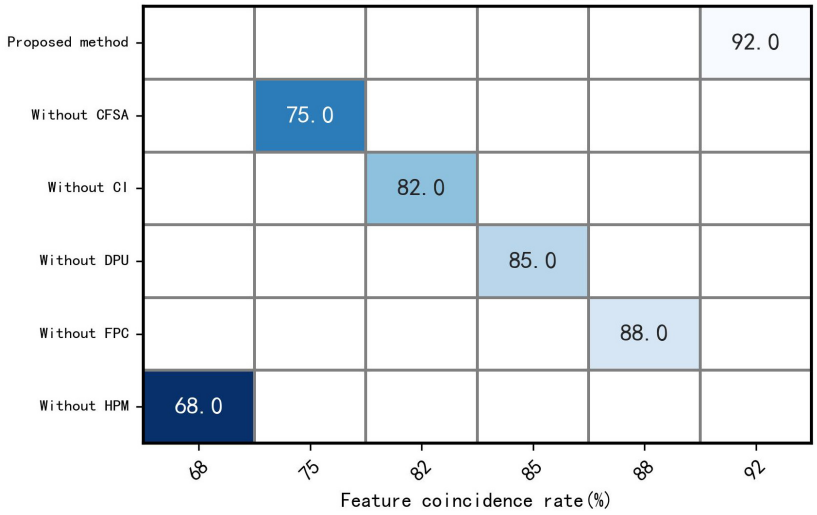


Figure 5. Map of the ablation experiments (feature coincidence rate across ablated modules).

Ablation analyses confirmed the indispensable role of each component: As shown in Figure 4, removing CI reduced the optimal F1-score by 4.5% and increased SHAP value variance by 0.02, validating its role in enhancing initial solution diversity to stabilize feature selection and reduce prediction volatility. Eliminating DPU delayed convergence by 10 iterations and lowered the F1-score by 3% with a trend clearly reflected in Figure 4, underscoring adaptive parameter tuning's role in balancing global exploration and local exploitation. Ablating FPC increased SHAP value variance to 0.05 (150% higher than the complete model; see Figure 4), demonstrating the penalty term's efficacy in eliminating redundant features and improving feature importance consistency. Removing CFSA caused a 6.7% F1 decline and a 17% drop in the feature coincidence rate, confirming that alignment mitigates domain shift. HPM ablation yielded the most severe degradation: no transfer reduced F1 by 8.9%, while full transfer decreased F1 by 6.7%, validating the "80% freezing + 20% fine-tuning" strategy. Collectively, these results highlight synergies between PSO optimization and cross-domain modules, reinforcing their critical role in enhancing small-sample EHM feature selection robustness.

The experimental results on the CWRU-bearing dataset demonstrate the effectiveness of the proposed model in small-sample feature selection for rolling bearing health management. However, it should be noted that the applicability of the model to other equipment

types and EHM data modalities requires further validation, which will be addressed in future research.

5. Conclusions

Addressing the challenge of small-sample feature selection in equipment health management, this paper proposed an improved model integrating dynamic PSO optimization and cross-domain TL. By introducing chaotic initialization and adaptive parameter adjustment strategies, the global search capability and convergence efficiency of feature selection were significantly enhanced. Compared to the traditional PSO algorithm, the proposed model converged 14 iterations earlier, and the feature overlap increased from 68% to 92%, effectively balancing model performance and feature sparsity. Through a hierarchical strategy of source domain to target domain feature space alignment, front-layer parameter freezing and back-layer fine-tuning, combined with a hybrid training loss function, the overfitting problem in small-sample scenarios was successfully mitigated. The F1-score of the target domain validation reached 0.89, which was 9% higher than that of the non-transfer method, while the recall rate for severe faults improved by 23.6%. Comparative experiments on several classifiers demonstrated that the selected 10-dimensional features exhibited good adaptability to mainstream models, such as XGBoost and RF. Notably, XGBoost achieved a recall rate of 0.89 for severe fault identification, verifying the cross-model generalization ability of the selected feature subset.

This study provides an effective solution for small-sample feature selection in equipment health management, but it also has certain limitations. The proposed model was only validated on the CWRU-bearing dataset—a widely used benchmark for rolling bearing fault diagnosis and health management. Although this dataset covers multiple working conditions (e.g., normal, inner ring fault, outer ring fault and rolling element fault) and load levels (0 HP, 1 HP, 2 HP and 3 HP), it primarily focuses on the vibration signal features of rolling bearings. The characteristics of equipment health-related data vary significantly across different types of equipment (e.g., gears, motors and hydraulic systems) and monitoring modalities (e.g., temperature, pressure and acoustic emission). Thus, the performance of the proposed dynamic PSO-optimized XGBoost–RFE model with cross-domain hierarchical transfer has not been verified on other EHM datasets with distinct data distributions, feature types or fault modes. This may restrict the direct generalization of the model to broader equipment health management scenarios beyond rolling bearing vibration monitoring.

The experimental results demonstrated that the proposed model significantly enhanced feature selection efficiency and classification performance in small-sample scenarios, providing an effective solution for the health management of new equipment during initial phases with insufficient data. Future research will focus on two key directions to address the aforementioned limitations and promote practical applications: First, regarding multi-dataset validation and generalization testing, the proposed model will be tested on diverse EHM datasets covering different equipment types and monitoring modalities. This will involve comparing the model's performance across datasets with varying characteristics—such as high-dimensional heterogeneous data, non-stationary signals and imbalanced fault sample distributions—to verify its robustness and generalizability. Second, multi-source heterogeneous data fusion and online real-time selection will integrate multi-modal data to construct more comprehensive health feature sets and develop an online adaptive feature selection mechanism based on the proposed model. This will enable real-time adjustment of feature selection strategies according to dynamic changes in equipment operating conditions, further supporting the full life-cycle health management of complex equipment.

Author Contributions: Conceptualization: Y.L., J.Z. and W.L.; methodology: Y.L., J.Z. and W.L.; formal analysis and data curation: Y.L., J.Z., W.L. and Y.H.; writing—original draft preparation: Y.L.; writing—review and editing: Y.L., J.Z., W.L. and Y.H.; supervision: Y.L., J.Z., W.L. and Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data supporting the reported results are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Li, X.Q. Research on Key Technologies of Complex Equipment Fault Prediction and Health Management. Ph.D. Thesis, Beijing Institute of Technology, Beijing, China, 2014.
2. Zhao, Z.A. Verification Technology of Fault Prediction Ability of Equipped PHM System. Ph.D. Thesis, National University of Defense Technology, Changsha, China, 2018.
3. Zhang, H.B.; He, Q.B. Tacholess bearing fault detection based on adaptive impulse extraction in the time domain under fluctuant speed. *Meas. Sci. Technol.* **2020**, *31*, 074004. [CrossRef]
4. Jiao, H.C.; Sun, W.L.; Wang, H.W.; Wan, X.J. Comprehensive Exploitation of Time- and Frequency-Domain Information for Bearing Fault Diagnosis on Imbalanced Datasets via Adaptive Wavelet-like Transform General Adversarial Network and Ensemble Learning. *Sensors* **2025**, *25*, 2328. [CrossRef]
5. Huo, J.Y.; Yang, J.W.; Yao, D.C.; Sun, R.T.; Hu, Z.S.; Chen, Z.H.; Gao, C. Bearing fault diagnosis under variable speed conditions on adaptive time frequency extraction mode decomposition. *Meas. Sci. Technol.* **2024**, *35*, 076102. [CrossRef]
6. Guo, L.; Liu, X.D.; Wei, D.T.; Zhu, P. Extraction Method of Equipment Health Characterization Parameters Based on Improved PCA. *Syst. Eng. Electron.* **2022**, *44*, 3275–3281. [CrossRef]
7. Diallo, A.F.; Patras, P. Deciphering Clusters with a Deterministic Measure of Clustering Tendency. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 1489–1501. [CrossRef]
8. Vahldiek, K.; Klawonn, F. Cluster-Centered Visualization Techniques for Fuzzy Clustering Results to Judge Single Clusters. *Appl. Sci.* **2024**, *14*, 1102. [CrossRef]
9. Yan, X.Y.; Nazmi, S.; Erol, B.A.; Homaifar, A.; Gebru, B.; Tunstel, E. An Efficient Unsupervised Feature Selection Procedure Through Feature Clustering. *Pattern Recognit. Lett.* **2020**, *131*, 277–284. [CrossRef]
10. Pandove, D.; Goel, S.; Rani, R. General Correlation Coefficient Based Agglomerative Clustering. *Clust. Comput.* **2019**, *22*, 553–583. [CrossRef]
11. Rahmat, F.; Zulkafli, Z.; Ishak, A.J.; Abdul Rahman, R.Z.; Stercke, S.D.; Buytaert, W.; Tahir, W.; Ab Rahman, J.; Ibrahim, S.; Ismail, M. Supervised Feature Selection Using Principal Component Analysis. *Knowl. Inf. Syst.* **2024**, *66*, 1955–1995. [CrossRef]
12. Atteia, G.; Alnashwan, R.; Hassan, M. Hybrid Feature-Learning-Based PSO-PCA Feature Engineering Approach for Blood Cancer Classification. *Diagnostics* **2023**, *13*, 2672. [CrossRef] [PubMed]
13. Zhao, M.X.; Jiang, H.; Chen, Q.S. Identification of Procymidone in Rapeseed Oils Based on Olfactory Visualization Technology. *Microchem. J.* **2023**, *193*, 109055. [CrossRef]
14. Zhao, M.X.; Jiang, H.; Chen, Q.S. Determination of Residual Levels of Procymidone in Rapeseed Oil Using Near-Infrared Spectroscopy Combined with Multivariate Analysis. *Infrared Phys. Technol.* **2023**, *133*, 104827. [CrossRef]
15. Saeys, Y.; Inza, I.; Larrañaga, P. A Review of Feature Selection Techniques in Bioinformatics. *Bioinformatics* **2007**, *23*, 2507–2517. [CrossRef]
16. Paulsson, N.; Larsson, E.; Winquist, F. Extraction and Selection of Parameters for Evaluation of Breath Alcohol Measurement with an Electronic Nose. *Sens. Actuators A Phys.* **2000**, *84*, 187–197. [CrossRef]
17. Jeong, Y.S.; Shin, K.S.; Jeong, M.K. An Evolutionary Algorithm with the Partial Sequential Forward Floating Search Mutation for Large-Scale Feature Selection Problems. *J. Oper. Res. Soc.* **2015**, *66*, 529–538. [CrossRef]
18. Deng, S.B.; Li, Y.L.; Wang, J.K.; Cao, R.T.; Li, M. A Feature-Thresholds Guided Genetic Algorithm Based on a Multi-Objective Feature Scoring Method for High-Dimensional Feature Selection. *Appl. Soft Comput.* **2023**, *148*, 110765. [CrossRef]
19. Pei, J.; Xu, L.; Huang, Y.T.; Jiao, Q.B.; Yang, M.Y.; Ma, D.; Jiang, S.; Li, H.; Li, Y.; Liu, S.; et al. A Two-Step Simulated Annealing Algorithm for Spectral Data Feature Extraction. *Sensors* **2023**, *23*, 893. [CrossRef] [PubMed]
20. Cho, J.H.; Kurup, P.U. Decision Tree Approach for Classification and Dimensionality Reduction of Electronic Nose Data. *Sens. Actuators B Chem.* **2011**, *160*, 542–548. [CrossRef]
21. Wang, K.; Ma, C.G.; Ao, R.; Liu, J.C.; Qiu, T.Y.; Zhang, L.J.; Xin, L.L.; Xin, J.Y.; Sun, Q.L. Valve Leakage Signal Feature Selection Method Based on XGBoost Model. *Chin. J. Valve* **2024**, *10*, 1227–1234. [CrossRef]

22. Zhang, B.; Zhang, Y.; Jiang, X.C. Feature Selection for Global Tropospheric Ozone Prediction Based on the BO-XGBoost-RFE Algorithm. *Sci. Rep.* **2022**, *12*, 9244. [CrossRef]
23. Chen, Z.Y.; Gryllias, K.; Li, W.H. Intelligent Fault Diagnosis for Rotary Machinery Using Transferable Convolutional Neural Network. *IEEE Trans. Ind. Inform.* **2020**, *16*, 339–349. [CrossRef]
24. Chen, H.T.; Luo, H.; Huang, B.; Jiang, B.; Kaynak, O. Transfer Learning-Motivated Intelligent Fault Diagnosis Designs: A Survey, Insights, and Perspectives. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 2969–2983. [CrossRef] [PubMed]
25. Wu, H.; Fan, Z.M.; Gan, L. Feature Transfer Learning for Fatigue Life Prediction of Additive Manufactured Metals with Small Samples. *Fatigue Fract. Eng. Mater. Struct.* **2025**, *48*, 467–486. [CrossRef]
26. He, Q.Q.; Siu, S.W.I.; Si, Y.W. Instance-Based Deep Transfer Learning with Attention for Stock Movement Prediction. *Appl. Intell.* **2023**, *53*, 6887–6908. [CrossRef]
27. Zhou, R.; Qiao, B.J.; Jiang, L.L.; Cheng, W.; Yang, X.Y.; Wang, Y.N.; Chen, X.F. A Model-Based Deep Learning Approach to Interpretable Impact Force Localization and Reconstruction. *Mech. Syst. Signal Process.* **2025**, *224*, 111977. [CrossRef]
28. Açikkar, M. Fast Grid Search: A Grid Search-Inspired Algorithm for Optimizing Hyperparameters of Support Vector Regression. *Turk. J. Electr. Eng. Comput. Sci.* **2024**, *32*, 68–92. [CrossRef]
29. Deligkaris, K. Particle Swarm Optimization and Random Search for Convolutional Neural Architecture Search. *IEEE Access* **2024**, *12*, 91229–91241. [CrossRef]
30. Pan, Q.S.; Zhang, C.; Wei, X.Y.; Wan, A.D.; Wei, Z.P. Assessment of MV XLPE Cable Aging State Based on PSO-XGBoost Algorithm. *Electr. Power Syst. Res.* **2023**, *221*, 109427. [CrossRef]
31. Wold, S.; Esbensen, K.; Geladi, P. Principal Component Analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [CrossRef]

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