

Article

Fault Prediction Modeling for High-Impact Recorders Based on IPSO-SVM

Linyu Li ^{1,2}, Wenbin You ^{1,2,*} and Yonghong Ding ^{1,2,*}

¹ The National Key Laboratory of Electronic Test Technology, North University of China, Taiyuan 030051, China; llybcys@163.com

² School of Electrical and Control Engineering, North University of China, Taiyuan 030051, China

* Correspondence: youwenbin@nuc.edu.cn (W.Y.); dingyonghong@nuc.edu.cn (Y.D.)

Abstract: The challenge in reusing high-impact recorders lies in developing an efficient and accurate failure prediction model under small-sample conditions. To address this issue, this study proposes an IPSO-SVM model. First, the particle swarms in the IPSO algorithm were grouped based on their exploration and exploitation functions, and dynamic inertia weight mechanisms were designed accordingly. The grouping ratio was dynamically adjusted during iterations to enhance optimization performance. Tests using benchmark functions verified that this approach improves convergence accuracy and stability compared to conventional PSO algorithms. Subsequently, the 5-fold cross-validation accuracy of the SVM model was used as the fitness value, and the IPSO algorithm was employed to optimize the penalty and kernel parameters of the SVM model. Trained on high-impact experimental data, the IPSO-SVM model achieved a prediction accuracy of 90.5%, outperforming the PSO-SVM model's 85%. These results demonstrate the potential of the IPSO-SVM model in addressing failure prediction challenges under small-sample conditions.

Keywords: fault prediction; high-impact recorder; particle swarm optimization algorithm (PSO); support vector machine (SVM)



Academic Editor: Alexandre Carvalho

Received: 11 December 2024

Revised: 24 January 2025

Accepted: 25 January 2025

Published: 27 January 2025

Citation: Li, L.; You, W.; Ding, Y. Fault Prediction Modeling for High-Impact Recorders Based on IPSO-SVM. *Appl. Sci.* **2025**, *15*, 1343. <https://doi.org/10.3390/app15031343>

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

High-impact recorders are crucial equipment in the development of new weapon systems. They capture critical data during weapon flight, such as acceleration, angular velocity, and pressure, enabling weapon developers to verify the operational status of the equipment and perform fault analysis [1,2]. In the past, after enduring a single high-impact test, it was difficult to determine whether the recorder could be reliably reused, so it was typically discarded. To reduce weapon development costs, there has been increasing interest in developing reusable high-impact recorders, making fault prediction a pressing challenge.

Currently, fault models for instruments and equipment are generally divided into physical models and data-driven models. Data-driven models have significant advantages over traditional physical models, particularly for complex and nonlinear systems, as they can reveal fault patterns and system behaviors through learning and analyzing historical operational data. Common data-driven models include neural networks, support vector machines (SVMs), and other methods [3,4]. In recent years, neural networks have been widely applied in fault prediction and health management. For instance, one study [5] proposed using spiking neural networks for fault prediction in syntactical time series. Another study [6] introduced a method combining particle swarm optimization (PSO) and backpropagation neural networks (BPNN-PSO) for fault diagnosis in photovoltaic array

systems. These studies illustrate that neural networks show good performance and wide applicability in fault prediction. However, training most neural networks requires large amounts of high-quality data, which is often limited during the development of many instruments. Under small-sample conditions, the performance of neural networks may not be as effective as desired. In such cases, SVM models may be a better option. One study [7] demonstrated the application of SVM for fault diagnosis in rotating machinery, highlighting its strong performance under small-sample conditions. Additionally, another study [8] noted that, with appropriate preprocessing methods, SVM could achieve excellent classification results in certain cases. Research [9] tested various methods, including PSO-SVM, HHO-SVM, and MFO-SVM, and confirmed their effectiveness in optimizing SVM models, achieving promising results across different aspects.

These findings suggest that SVM models perform well for small-sample fault prediction. However, their performance is influenced by the selection of penalty and kernel parameters. In traditional SVM models, these parameters are often determined based on empirical values, leading to unstable performance. To improve fault prediction accuracy, it is necessary to explore suitable optimization algorithms for determining SVM parameters. The particle swarm optimization (PSO) algorithm is a commonly used function optimizer that can improve SVM fault recognition accuracy to some extent. However, it is prone to premature convergence and getting trapped in local optima, indicating room for enhancement [10].

In recent years, many advanced optimization algorithms have been proposed and applied to solve complex problems. For instance, [11] reviews the development of the Whale Optimization Algorithm (WOA), analyzing 59 improved variants and 57 hybrid versions, and suggests effective strategies to enhance exploration capabilities and optimization performance. Reference [12] introduces a novel exploration strategy for the YUKI algorithm, which dynamically adjusts exploration and exploitation strategies and integrates gradient descent to improve performance in topology optimization, significantly enhancing its exploration ability. These studies provide valuable insights and references for improving SVM parameter optimization.

In this paper, we propose an improved PSO-SVM model (hereinafter referred to as IPSO-SVM) to address the challenge of fault prediction for high-impact recorders under small-sample conditions. First, we enhanced the standard PSO algorithm by grouping particles based on their exploration and exploitation functions and applying dynamic inertia weight mechanisms, with the grouping ratio adjusted in real-time during iterations. The performance of the IPSO algorithm was validated using the CEC2022 benchmark functions, demonstrating improved convergence accuracy and stability compared to conventional PSO algorithms. Next, ten high-impact recorders were used in impact experiments, and the experimental data were split and expanded to create training and test datasets. During the training process, the IPSO algorithm was used to optimize the SVM penalty and kernel parameters. Finally, the model's accuracy was validated using the test dataset.

2. Materials

2.1. SVM Classifier

The support vector machine (SVM) is a supervised learning model whose fundamental principle is to identify a decision boundary that maximizes the classification margin. This ensures that data points from different classes are correctly classified while positioning the boundary as far as possible from the nearest data points, known as support vectors. Maximizing the classification margin improves the model's generalization ability, resulting in better performance on unseen data [13].

For nonlinear problems, the SVM model introduces a kernel function that maps the original input into a higher-dimensional space. In this space, the decision boundary is determined and used to classify nonlinear data [14]. The SVM model function is expressed as shown in Equation (1).

$$f(x) = \text{sign}\left(\sum_{i=1}^n \alpha_i y_i K(x_i, x) + b\right) \quad (1)$$

where:

- $f(x)$: The output of the model, representing the predicted class label for the input x .
- α_i : The Lagrange multipliers, which represent the weights of the support vectors in the model.
- y_i : The class label of the i -th support vector, indicating its corresponding category.
- $K(x_i, x)$: The kernel function, which calculates the similarity between the support vector x_i and the input x in the transformed high-dimensional space.
- b : The bias term, which helps to adjust the decision boundary away from the origin in the high-dimensional space.

In the SVM model, the objective is to identify the optimal decision boundary that maximizes the classification margin while appropriately penalizing misclassified samples. To achieve this, the SVM adopts a dual problem formulation, converting the optimization problem using the Lagrange multiplier method [15]. The objective function of the dual problem, along with its constraints, is expressed as shown in Equation (2).

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^n \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C \end{cases} \end{aligned} \quad (2)$$

The SVM model typically requires the determination of the following functions:

1. **Penalty Parameter:** A larger value of this parameter forces the model to fit the training data more strictly, which can lead to overfitting. While this may result in better performance on the training data, it can potentially degrade performance on the test data. Conversely, if the penalty parameter is too small, it becomes more lenient toward misclassifications, resulting in a larger margin that may lead to underfitting of the SVM model.
2. **Kernel Function and Kernel Parameters:** The kernel function utilized in this paper is the RBF (Radial Basis Function) kernel. The parameter associated with this kernel governs the influence range of data points; a smaller value enhances the generalization performance of the SVM model, while a larger value may degrade its generalization capability [16].

2.2. IPSO Algorithm

To enhance the recognition accuracy of the SVM fault prediction model, it is essential to optimize its penalty parameter and kernel function parameters. This paper introduces an enhancement to the PSO algorithm.

The standard PSO algorithm initializes multiple particles within a D-dimensional search space to find the optimal solution. Each particle begins at a random position with a random velocity and iteratively updates its velocity and position according to Equations (3) and (4) [17].

$$v_{id}(t+1) = wv_{id}(t) + c_1 r_1 (Pbest_{id}(t) - x_{id}(t)) + c_2 r_2 (Gbest_d(t) - x_{id}(t)) \quad (3)$$

$$x_{id}(t + 1) = x_{id}(t) + v_{id}(t + 1) \tag{4}$$

To tackle the issues of premature convergence and being trapped in local optima, which are common in the early stages of the standard PSO algorithm, this paper introduces the Improved Particle Swarm Optimization (IPSO) algorithm, which incorporates the following modifications to the particle swarm optimization process:

1. The particle swarm is divided into an exploration group and an exploitation group. The role of the exploration group is to enhance the algorithm’s global search capability. Particles in this group are less influenced by the global best position and can accept inferior solutions with a certain probability, allowing them to escape local optima. In contrast, the exploitation group aims to converge quickly. Particles in this group are more strongly influenced by the global best position and remain within the vicinity of the local optimum.

A dynamic grouping method is designed for the particle grouping ratio. In the early stages of the search, a larger number of particles are allocated to global exploration. As the search progresses, more particles gradually transition to the exploitation group. As shown in Equation (5), where $k \in (0, 1)$, its function is to control the timing of when all particles in the exploration group transition to the exploitation group. When $s_i = -1$, the particle belongs to the exploration group; when $s_i = 1$, it belongs to the exploitation group. If $t > kT$, all particles transition to the exploitation group.

$$s_i = \begin{cases} -1 & n_i - n_{\max}(L_1 + L_2 \times (\frac{t}{kT})^j) > 0 \\ 1 & n_i - n_{\max}(L_1 + L_2 \times (\frac{t}{kT})^j) \leq 0 \end{cases} \tag{5}$$

In this equation, L_1 and L_2 control the initial ratio of the two groups of particles, and their sum equals 1. In this paper, the values of L_1 and L_2 are 0.2 and 0.8, respectively. The parameter j is a positive number that regulates the speed at which the exploration group transitions to the exploitation group, with a value of 1 in this study.

2. For the exploration group of particles, the introduction of dynamic inertia weights aims to minimize their influence from the global best position in the early stages. As iterations progress, these weights gradually lead to convergence at the global optimum. The inertia weight is expressed in Equation (6).

$$w(t) = (w_{\max} - w_{\min})\left(\frac{T - t}{T}\right)^{0.5} + w_{\min} \tag{6}$$

For the exploitation group of particles, a larger inertia weight facilitates rapid convergence, and the dynamic inertia weight increases quickly. The inertia weight formula for the exploitation group is defined in Equation (7).

$$w(t) = (w_{\max} - w_{\min})(1 - 1/(1 + e^{-(t-T \times 0.2)})) + w_{\min} \tag{7}$$

3. One of the drawbacks of PSO is its tendency to become trapped in local optima. To address this issue, inspired by the simulated annealing algorithm, the algorithm accepts a new position generated during iterations with a poor fitness value with a certain probability. This approach allows the algorithm to escape local optima. The acceptance probability is defined in Equation (8).

$$p = \begin{cases} 1 & \text{if } f(x_{\text{new}}) < f(x_{\text{current}}) \\ e^{-\frac{f(x_{\text{new}}) - f(x_{\text{current}})}{T}} & \text{if } f(x_{\text{new}}) \geq f(x_{\text{current}}) \end{cases} \tag{8}$$

2.3. Behavior of IPSO

To verify whether the grouped particles in the IPSO algorithm perform as intended—i.e., the exploration group conducts a global search while the exploitation group focuses near the minimum value—the variance of inter-particle distances was used as the evaluation metric. A higher variance indicates a broader search range, while a lower variance suggests a more focused and localized search.

$$f = \sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^2 + \left(\sum_{i=1}^n 0.5ix_i\right)^4 \tag{9}$$

The first three experiments utilized the standard PSO algorithm with inertia weights set to a constant value of 0.5, as well as Formulas (6) and (7). The fourth experiment examined the variance values of particles under the combined behavior of the IPSO algorithm. The Zakharov function, shown in Formula (9), was selected as the benchmark function to find its minimum value under the initial conditions of 3 dimensions, 30 particles, and 100 iterations. The corresponding data is summarized in Table 1.

Table 1. Average Variance of Inter-Particle Distances Across Different Iterations.

Iteration	Standard PSO	Exploration Group	Exploitation Group	IPSO
1–25	1417.24	4043.52	1455.71	3467.72
26–50	1052.74	4418.87	896.49	1859.96
50–75	658.23	3921.64	508.65	895.33
76–100	372.30	2912.25	264.68	366.09

Analyzing these data, the following conclusions can be drawn:

- The exploration group achieved the highest variance throughout the iterations, demonstrating its capability to perform broad global searches.
- The exploitation group maintained significantly lower variance values, reflecting its concentration near the optimal solution.
- The standard PSO showed a gradual decrease in variance, indicating a more balanced yet less specialized search approach.

This suggests that the independent design of the exploration and exploitation groups has met expectations. However, deep exploration near the optimal solution in the early stages is of limited use, as the global best position is likely to change rapidly. Similarly, extensive searches in the later stages have a lower probability of finding better solutions and are less conducive to convergence. To address this, the IPSO algorithm introduces a dynamic grouping strategy, allowing particles in the exploration group to gradually transition to the exploitation group as iterations progress. This adjustment effectively eliminates redundant global searches in the later stages. As a result, during iterations 76 to 100, the variance levels of IPSO closely align with those of PSO. This demonstrates that IPSO successfully balances the behaviors of the two groups of particles. Furthermore, the broader exploration in the early iterations enables IPSO to outperform PSO, yielding better search results. This improvement will be further discussed in the next section.

2.4. Performance of IPSO

To validate the performance of the IPSO algorithm, the CEC2022 benchmark test functions were used for performance evaluation. In the CEC2022 set, Function 1 is a Unimodal Function, Functions 2–5 are Basic Functions, Functions 6–8 are Hybrid Functions,

and Functions 9–12 are Composition Functions [18]. For this study, Functions 1, 2, 7, and 9 were selected as representative test cases, as shown in Table 2.

Table 2. CEC2022 benchmark test functions.

Sequence	Function
Func1	$f_1 = \sum_{i=1}^n x_i^2 + (\sum_{i=1}^n 0.5ix_i)^2 + (\sum_{i=1}^n 0.5ix_i)^4$
Func2	$f_2 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
Func7	$f_7 = \left (\sum_{i=1}^n x_i^2)^2 - (\sum_{i=1}^n x_i)^2 \right ^{0.5} + (0.5\sum_{i=1}^n x_i^2 + \sum_{i=1}^n x_i) / n + 0.5$
Func9	$f_9 = \prod_{i=1}^n \left(1 + i \sum_{j=1}^n \frac{2^j x_i - \text{round}(2^j x_i)}{2^j} \right)^{\frac{10}{n^{1.2}}} - 1$

The IPSO algorithm and the PSO algorithm were run on each test function, with the optimization constraints as shown in Table 3.

Table 3. Optimization Constraints for IPSO and PSO Algorithms.

Sequence	Dimension	Number of Particles	Number of Iterations	Bounds
Func1	20	30	10	[−100, 100]
Func2	20	30	100	[−100, 100]
Func7	20	30	100	[−100, 100]
Func9	20	30	500	[−100, 100]

The results were plotted as fitness curves, as shown in Figure 1. It can be observed that the IPSO algorithm outperforms the PSO algorithm in terms of both convergence speed and accuracy.

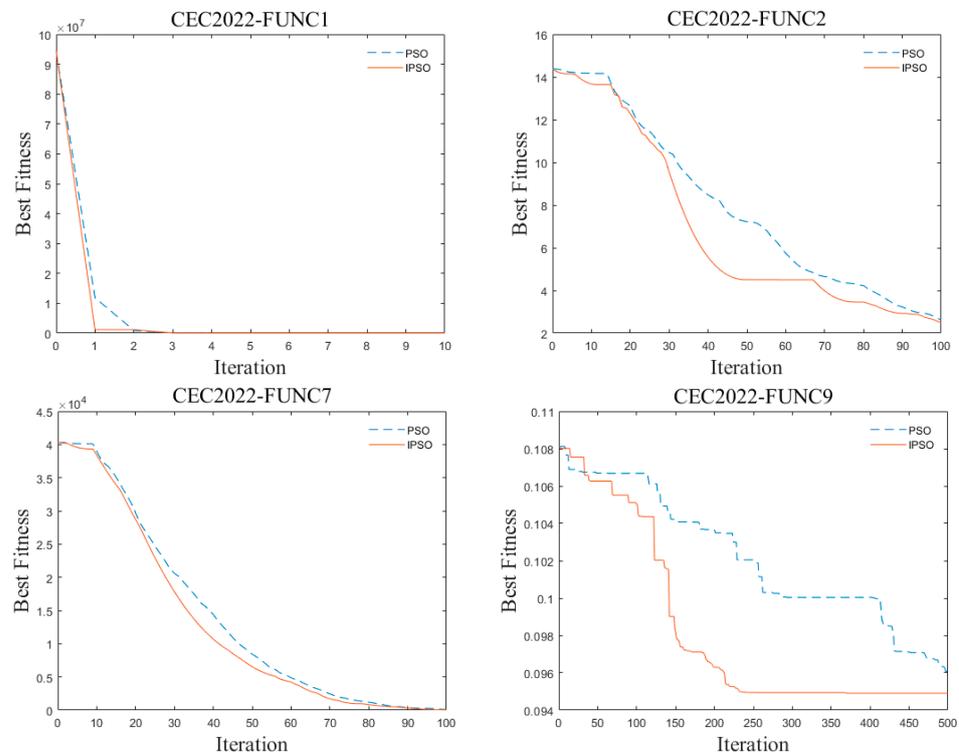


Figure 1. Fitness curve.

However, both algorithms exhibit a degree of randomness during the optimization process. To obtain more reliable data, both algorithms were executed 30 times on each test

function, and their performance was compared using optimal values, worst values, means, and variances. The test results are summarized in Table 4. Compared to the traditional PSO algorithm, the IPSO algorithm shows improvements in both the optimal and average values, with a significant reduction in variance, indicating higher convergence accuracy and greater stability.

Table 4. Optimization Results of Test Functions.

Function	Algorithm	Optimal Values	Worst Values	Means	Variances
Func1	PSO	3.417	45.10	15.52	107.2
	IPSO	2.889	27.75	10.40	40.00
Func2	PSO	60.33	3103	681.6	3.52×10^5
	IPSO	14.61	164.2	16.64	849.9
Func7	PSO	1.017	4.602	2.198	0.572
	IPSO	0.4364	1.011	0.6271	0.0213
Func9	PSO	0.0901	0.1010	0.0960	6.84×10^{-6}
	IPSO	0.0890	0.0993	0.0946	4.47×10^{-7}

2.5. Prediction Model Construction

The impact signals collected multiple times from the high-impact recorder exhibit high dimensionality and non-linearity. Furthermore, the acquired experimental data are limited, resulting in a small sample problem. SVM is particularly effective in addressing challenges related to small samples, non-linearity, and high-dimensional pattern recognition. Therefore, this paper utilizes SVM to construct the fault prediction model. To enhance the performance of the SVM-based fault prediction model, the IPSO algorithm is employed for parameter selection of the SVM model. Figure 2 illustrates the flowchart for constructing the IPSO-optimized SVM fault prediction model, with the specific steps outlined as follows:

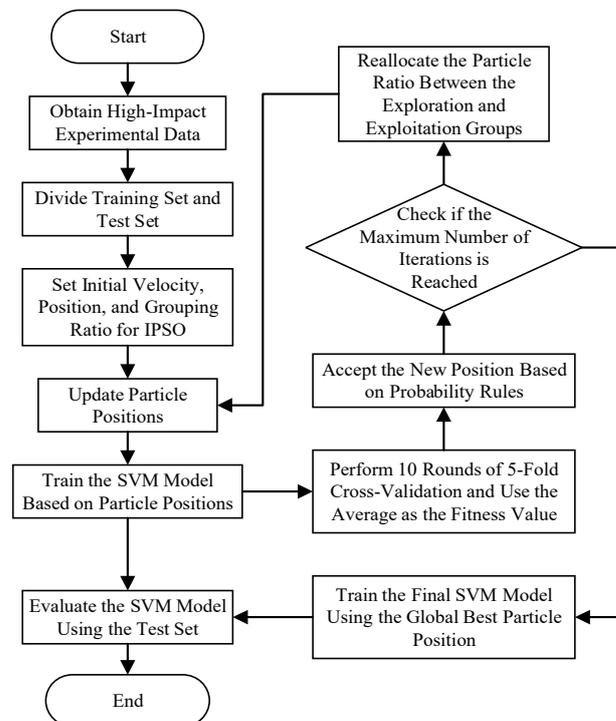


Figure 2. Model Training Flowchart.

1. Divide the collected data that caused damage to the high-impact recorder into training and testing datasets.
2. Initialize the particle positions and velocities for the IPSO algorithm, and group the particles based on the initial ratio.
3. Update the particle positions, which represent the parameters to be determined for the SVM. Train the SVM model using the parameters from each group with the training dataset.
4. Conduct 10 rounds of 5-fold cross-validation on the trained SVM model, using the average as the fitness value.
5. Compare the fitness value of the current iteration with that of the previous iteration. If the current fitness value is better, accept the new position; if the fitness value is worse but the particle belongs to the exploration group, accept the position with a certain probability according to the principle outlined in Equation (8).
6. If the maximum number of iterations has not been reached, some particles from the exploration group will join the exploitation group, and the process returns to Step 3.
7. If the maximum number of iterations has been reached, use the global best particle position as the output, which serve as the final parameters for the SVM model. The final SVM model is obtained by training with these parameters and the training dataset.
8. Evaluate various metrics of the SVM model using the testing dataset to draw experimental conclusions.

3. Experiments and Results

3.1. Data Collection

To construct the IPSO-SVM model for fault prediction in high-impact recorders, high-impact experiments were conducted. The specific steps for experimental design and data collection are as follows:

3.1.1. Determining the Experimental Range

Before the experiments, simulations of the PCB's response under high-impact conditions were conducted using Ansys and Is-Dyna. The purpose of these simulations was to analyze the stress distribution and dynamic response of the recorder across different impact acceleration ranges to determine a reasonable range for the experimental accelerations.

Ansys: The finite element method was used to construct the structural model of the recorder, simulating the stress distribution and strain under various impact acceleration conditions. The simulation identified potential weak points and vulnerable areas in the recorder, providing a design basis for the experiments.

Is-Dyna: The dynamic response of the recorder under high-impact loading was simulated, predicting its failure behavior within the acceleration range of 10,000 g to 100,000 g. The Is-Dyna simulation further calibrated the Ansys analysis results, ultimately determining that controlling the experimental range between 20,000 g and 90,000 g was more reasonable.

3.1.2. Connecting Equipment and Sensors

In the experiment, a high-impact acceleration sensor with the model number M350D02 (PCB Piezotronics, Depew, NY, USA) was used. The main parameters of the sensor are shown in Table 5.

After connecting the data recorder to the sensor, it was installed on the high-impact testing platform, as shown in Figure 3a. Acceleration curves were then collected, as shown in Figure 3b.

Table 5. Sensor Parameters.

Parameter	Specification
Model number	M350D02
Sensitivity	0.05 mV/g
Measurement Range	±100,000 g
Frequency Range	4 to 10,000 Hz
Weight	5.5 g

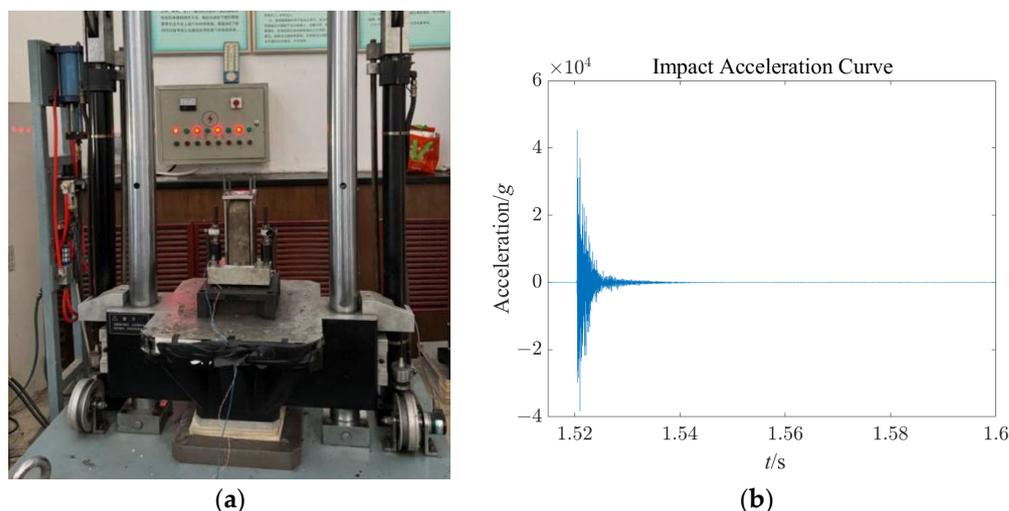


Figure 3. (a) Shock Test Table Model; (b) Acceleration Curves.

3.2. Experimental Grouping and Data Collection

A total of 10 experimental setups were used, with each setup undergoing multiple impact tests on the testing platform until the equipment was damaged. After each experiment, the equipment status was checked, and acceleration curves were recorded. Multiple sets of acceleration curves collected from each setup were used as training data. To ensure data diversity and representativeness, the following grouping design was implemented:

- 3 setups with incrementally increasing acceleration, but with different step sizes.
- 3 setups with relatively constant peak acceleration values.
- 4 setups with randomly varying acceleration within the range of 20,000 g to 90,000 g.

Acceleration peaks under different impact conditions were recorded, as shown in Table 6.

Table 6. Experimental Peak Acceleration.

Experiment Number	Peak Acceleration Value (Unit: g)
01	[24,439, 38,164, 50,529, 61,137, 72,144, 85,871]
02	[32,518, 41,486, 52,299, 62,395, 74,480, 82,090]
03	[46,702, 51,136, 58,175, 64,463, 71,025, 78,388, 78,388]
04	[63,097, 61,659, 60,010, 61,211, 63,038, 60,193, 60,078, 62,297, 61,125, 61,269]
05	[65,681, 66,628, 65,624, 68,690, 71,801, 70,041, 69,895, 70,766]
06	[79,280, 75,419, 78,034, 81,033, 76,744]
07	[31,368, 73,325, 79,391, 35,725, 64,369, 88,706]
08	[48,370, 51,866, 76,331, 31,079, 58,324, 23,794, 40,863, 57,906, 39,087, 87,732]
09	[37,443, 56,419, 33,818, 79,596, 27,284, 76,450, 47,402, 88,521]
10	[25,139, 65,858, 34,232, 71,183, 46,534, 59,143, 86,526]

3.3. Data Processing

3.3.1. Data Labeling

To construct the dataset for the SVM, experimental data were split into two categories: “damaged” and “undamaged”, with labels 1 and 0, respectively. Based on the physical definition of the data, two types of key vectors were defined:

Critical Vectors:

Critical vectors are the acceleration data when the recorder transitions from an “undamaged” to a “damaged” state. For example, in the data shown in Table 6 (Data 01), the complete vector [24,439, 38,164, 50,529, 61,137, 72,144, 85,871] is labeled as “damaged” (label 1), while the vector obtained by removing the last data point, [24,439, 38,164, 50,529, 61,137, 72,144], is labeled as “undamaged” (label 0). Extracting critical vectors is crucial for the accuracy of the model because they contain key information about the device in its extreme state, making it easier for SVM models to identify them as support vectors.

Non-Critical Vectors:

Non-critical vectors are data points where the recorder is still in an “undamaged” state and can withstand more than one impact. For instance, from Table 6 (Data 01), the complete data [24,439, 38,164, 50,529, 61,137, 72,144, 85,871] can be used to extract shorter vectors, like [24,439], [24,439, 38,164], [24,439, 38,164, 50,529], and [24,439, 38,164, 50,529, 61,137], all of which are non-critical vectors and are labeled as “undamaged” (label 0). These vectors provide additional training samples, helping the model better learn the characteristics of the “undamaged” state.

As a result, from the 10 sets of experimental data in Table 6, a total of 20 critical vectors and 53 non-critical vectors were extracted.

3.3.2. Data Length Normalization and Expansion

Due to inconsistencies in vector lengths among the experimental data, all vectors were standardized to a length of 10 (using the longest dimension as the standard) to facilitate SVM model processing. For the missing parts, zeros were used as padding. The zero values indicate that the recorder was damaged, and no subsequent experiments were conducted, effectively skipping a particular experiment. It can be inferred that varying the position of the zero-padding does not alter the experimental results. However, the position of the zero elements can affect the SVM training outcomes. This allows for data expansion, as illustrated by the transformation, shown in Table 6 (Data 01), to the expanded format in Table 7.

Table 7. Example of Expanded Data.

Serial Number	Data Vector
01-01	[0, 0, 0, 0, 24,439, 38,164, 50,529, 61,137, 72,144, 85,871]
01-02	[0, 0, 0, 24,439, 0, 38,164, 50,529, 61,137, 72,144, 85,871]
01-03	[0, 0, 0, 24,439, 38,164, 0, 50,529, 61,137, 72,144, 85,871]
01-04	[0, 0, 0, 24,439, 38,164, 50,529, 0, 61,137, 72,144, 85,871]
01-05	[0, 0, 0, 24,439, 38,164, 50,529, 61,137, 0, 72,144, 85,871]
...	...

3.3.3. Composition of the Dataset

After the above processing, all critical vectors were expanded, resulting in a total of 2842 expanded critical vectors. From each dataset of critical vectors, 25 expanded vectors were randomly selected, totaling 500 expanded critical vectors for training. Additionally, since non-critical vectors are less likely to become support vectors in SVM training and excessive data can increase training time, 53 non-critical vectors were directly added to the training set. As a result, the training set contained 553 vectors in total. Finally, 200 expanded critical vectors were randomly selected from the remaining dataset to form the test set.

3.4. Results

Training vectors from the training set were used to train both the IPSO-SVM model and the PSO-SVM model. During the training process, the fitness values were recorded, and the fitness curve was plotted as shown in Figure 4. This demonstrates that, for the specific scenario of high-impact data, the IPSO algorithm outperforms the traditional PSO algorithm in terms of performance.

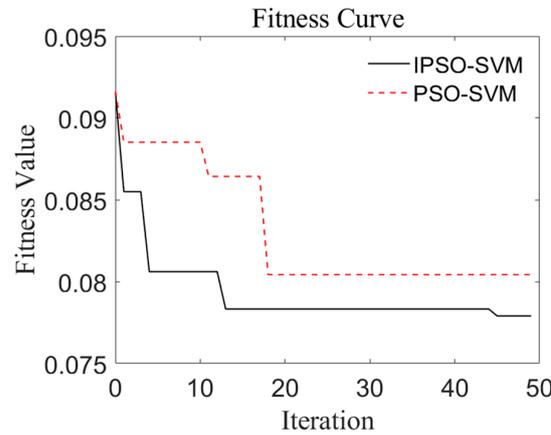


Figure 4. Fitness Curves.

Use the trained model to make predictions on the test set and record the number of correct predictions, as shown in Table 8.

Table 8. Prediction Results of the Test Set.

Model	Number of Test Samples	Actual Label	Number of Correct Predictions
IPSO-SVM	100	1	85
	100	0	96
PSO-SVM	100	1	81
	100	0	89

For SVM models, common evaluation metrics include accuracy, recall, F1 score, and precision. Based on the prediction results, these metrics were calculated for both models, as shown in Table 9.

Table 9. Model Validation Metrics.

Model	Accuracy	Precision	Recall	F1 Score
IPSO-SVM	90.5%	95.5%	85.0%	89.9%
PSO-SVM	85.0%	88%	81.0%	84.3%

The experimental results indicate that the IPSO-SVM model outperforms the traditional PSO-SVM model in terms of precision, accuracy, and recall, demonstrating superior performance in high-impact experiments. These findings further validate the potential of the IPSO-SVM model under small-sample conditions, proving that it can effectively enhance the accuracy and stability of fault prediction for high-impact equipment. Therefore, the IPSO-SVM model provides a valuable technical foundation for the reliability assessment of high-impact equipment and offers theoretical and practical guidance for implementing health monitoring and maintenance decisions in real-world applications.

3.5. Comparison with Other Models

In recent years, deep learning and other artificial intelligence methods have made significant advancements in fault prediction. Long Short-Term Memory (LSTM), a specialized type of Recurrent Neural Network (RNN), has been widely used in fault prediction due to its ability to capture long-term dependencies in time-series data. In this study, an LSTM-based fault prediction model was constructed to evaluate its performance in scenarios involving small samples, high dimensionality, and nonlinearity [19]. The basic parameters of the LSTM model are shown in Table 10:

Table 10. The basic parameters of the LSTM model.

Parameters	Value
Number of hidden units	128
Activation function	ReLU
Optimization algorithm	Adam
Learning rate	0.001
Batch size	32
Training epochs	50

Additionally, to provide a more comprehensive comparison, a commonly used machine learning algorithm—Random Forest—was selected as another baseline [20]. The parameters of the Random Forest model are shown in Table 11:

Table 11. The parameters of the Random Forest model.

Parameters	Value
Number of decision trees	100
Maximum depth	10
Minimum samples for split	2

Prediction results are shown in Table 12:

Table 12. Prediction Results of the IPSO-SVM model and other models.

Model	Accuracy	Precision	Recall	F1 Score
IPSO-SVM	90.5%	95.5%	85.0%	89.9%
LSTM	89.0%	91.5%	86.0%	88.7%
Random Forest	83.5%	86.0%	80.0%	82.9%

The data indicate that IPSO-SVM outperforms LSTM and Random Forest in fault prediction tasks. IPSO-SVM achieved the highest accuracy (90.5%), precision (95.5%), and F1 score (89.9%), followed by LSTM, which, despite having a higher recall rate (86%), had slightly lower precision and F1 score. Random Forest exhibited the lowest performance, with an accuracy of 83.5% and precision of 86.0%. These results suggest that IPSO-SVM is more effective than LSTM and Random Forest in handling small samples and nonlinear data.

4. Discussion

This study introduced the IPSO-SVM model to address the challenge of fault prediction for high-impact recorders under small sample conditions. By improving the Particle Swarm Optimization (PSO) algorithm through particle grouping, dynamic inertia weight mechanisms, and probabilistic acceptance of inferior solutions, the IPSO algorithm made significant progress in addressing the issues of premature convergence and local optima inherent in traditional PSO algorithms. Experimental and simulation data demonstrated that

the prediction accuracy of the IPSO-SVM model reached 90.5%, significantly outperforming the traditional PSO-SVM model.

The implications of this study are substantial. Firstly, it offers an effective technical solution to enhance the reusability of high-impact recorders, reducing potential costs associated with repeated use of the equipment. The IPSO-SVM model can also play a critical role in fault prediction tasks where the data are limited, high-dimensional, and nonlinear, making it particularly suitable for industrial applications where data are scarce or expensive to obtain.

In addition to the high-impact recorder applications, the IPSO-SVM model shows considerable promise in a variety of other industrial contexts. For instance, devices like acceleration recorders and free-field shock wave pressure recorders typically operate in extreme environments, with unpredictable reliability after each use. By collecting a small sample of historical operational data, the IPSO-SVM model can predict the future operational conditions of these devices, thus reducing the costs associated with frequent device replacements or failures caused by multiple uses. This is particularly valuable in industries, such as aerospace, defense, and energy, where equipment downtime or failure can be costly and detrimental.

Moreover, the IPSO-SVM model's effectiveness extends beyond high-impact equipment. It can also be applied in medical device development, where certain high-cost equipment cannot be used to gather large datasets. In scenarios involving small sample sizes, such as with costly medical instruments, the IPSO-SVM model can provide reliable fault predictions, helping to save on development costs and ensuring the accuracy of the devices without needing extensive testing datasets. This demonstrates the model's flexibility and potential for a wide range of applications, from high-end manufacturing to healthcare technology.

Despite these advancements, the study does acknowledge several limitations that offer opportunities for future research.

Limitations of the Model Research:

1. Limited dataset size: Future research could explore the robustness of the model by incorporating more diverse samples or different types of failure data.
2. Choice of optimization algorithm: Although the IPSO algorithm performed well in this study, other optimization algorithms (e.g., genetic algorithms or ant colony optimization) may perform better in different scenarios. Further comparative studies should be considered.

Analysis of Fault Prediction for High-Impact Recorders:

1. Data analysis results: Analysis of the data reveals that the model's success rate in predicting label 0 (undamaged) is higher than that for predicting label 1 (damaged). This may indicate that there is room for improvement in data collection methods and grouping structures during the experiment, warranting further exploration.
2. Choice of training data: This study only used the peak value of impact acceleration for training. Whether including additional data, such as bandwidth and frequency could lead to better prediction results still needs to be investigated.
3. Dimensionality issues: The data used in this experiment had 10 dimensions. Under higher-impact conditions, the data length may exceed this dimension, leading to longer training times and potentially impacting prediction results. To address this, dimensionality reduction techniques, such as Principal Component Analysis (PCA), could be considered. However, the specific impact of these techniques on prediction results is difficult to estimate and warrants further research.

5. Conclusions

This study proposes the IPSO-SVM model for fault prediction in high-impact recorders, addressing the challenge of limited sample sizes. The IPSO-SVM model achieved a prediction accuracy of 90.5%, outperforming the traditional PSO-SVM model by 5.5%. This improvement is attributed to the optimizations introduced in the IPSO algorithm, including particle grouping and dynamic inertia weights, which effectively address issues like premature convergence and local optima.

In addition to outperforming PSO-SVM, the IPSO-SVM model was also compared with other advanced models, including LSTM and Random Forest. The IPSO-SVM model showed superior performance, achieving a prediction accuracy that was 1.5% higher than LSTM and 7% higher than Random Forest in the fault prediction task. These comparisons demonstrate the IPSO-SVM model's ability to perform well with limited data, offering a more reliable prediction solution compared to traditional machine learning models.

The IPSO-SVM model also demonstrated its ability to handle high-dimensional data with limited samples, making it a valuable tool for predicting failures in high-impact equipment. The model's high accuracy ensures more reliable operation of equipment, reducing costs associated with frequent replacements and downtime.

Furthermore, this model has significant potential for application in other industrial fields, such as aerospace, military, and medical device development, where small datasets and high reliability are common challenges. For instance, it can be used to predict faults in devices like shock wave pressure recorders, thus enhancing reliability and reducing operational costs.

In conclusion, the IPSO-SVM model offers a promising solution for fault prediction in high-impact recorders, demonstrating improved accuracy and practical applicability in industries with limited data and high reliability demands.

Author Contributions: Conceptualization, L.L. and W.Y.; methodology, L.L. and Y.D.; software, L.L. and Y.D.; validation, L.L. and W.Y.; formal analysis, L.L.; investigation, L.L.; resources, W.Y.; data curation, L.L.; writing—original draft preparation, L.L.; writing—review and editing, L.L. and W.Y.; visualization, L.L. and W.Y.; supervision, L.L.; project administration, W.Y.; funding acquisition, W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 61701445 and the “Fundamental-Research-Program” of Shanxi Province, grant number 20210302124200.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

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

References

1. Jiang, W.C.; Lu, Y.G.; Zhao, J.Y. Researching the Influence of Preload on Vibration Characteristics in the Ballistic Recorder Vibration Damping System. *Shock Vib.* **2024**, *30*, 5868224. [[CrossRef](#)]
2. Sun, S.T.; Xv, J.L.; Wang, W. Shock Wave Measurement and Time Difference of Arrival Analysis for Impact Point Positioning. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 9600113. [[CrossRef](#)]
3. Pang, J. Adaptive fault prediction and maintenance in production lines using deep learning. *Int. J. Simul. Model.* **2023**, *22*, 734–745. [[CrossRef](#)]
4. Wang, H.Q.; Cai, Y.N. A Fast Small-Sample Modeling Method for Precision Inertial Systems Fault Prediction and Quantitative Anomaly Measurement. *CMES-Comput. Model. Eng. Sci.* **2022**, *130*, 187–203. [[CrossRef](#)]
5. Souza de Abreu, R.; Silva, I.; Nunes, Y.T.; Moiola, R.C.; Guedes, L.A. Advancing Fault Prediction: A Comparative Study between LSTM and Spiking Neural Networks. *Processes* **2023**, *11*, 2772. [[CrossRef](#)]

6. Eldeghady, G.S.; Kamal, H.A.; Hassan, M. Fault diagnosis for PV system using a deep learning optimized via PSO heuristic combination technique. *Electr. Eng.* **2023**, *105*, 2287–2301. [[CrossRef](#)]
7. Shi, P.; Liang, K.; Han, D.; Zhang, Y. A novel intelligent fault diagnosis method of rotating machinery based on deep learning and PSO-SVM. *J. Vibroeng.* **2017**, *19*, 5932–5946. [[CrossRef](#)]
8. Marić, D.; Duspara, M.; Šolić, T.; Samardžić, I. Application of SVM Models for Classification of Welded Joints. *Teh. Vjesnik-Tech. Gaz.* **2019**, *26*, 533–538.
9. Zhou, J.; Yang, P.; Peng, P.; Khandelwal, M.; Qiu, Y. Performance Evaluation of Rockburst Prediction Based on PSO-SVM, HHO-SVM, and MFO-SVM Hybrid Models. *Min. Metall. Explor.* **2023**, *40*, 617–635. [[CrossRef](#)]
10. Samantaray, S.; Sahoo, A.; Agnihotri, A. Prediction of Flood Discharge Using Hybrid PSO-SVM Algorithm in Barak River Basin. *Methodsx* **2023**, *10*, 102060. [[CrossRef](#)] [[PubMed](#)]
11. Nadimi-Shahraki, M.H.; Zamani, H.; Asghari Varzaneh, Z.; Mirjalili, S. A Systematic Review of the Whale Optimization Algorithm: Theoretical Foundation, Improvements, and Hybridizations. *Arch. Comput. Methods Eng.* **2023**, *30*, 4113–4159. [[CrossRef](#)] [[PubMed](#)]
12. Benaissa, B.; Kobayashi, M.; Al Ali, M.; Khatir, S.; Shimoda, M. A novel exploration strategy for the YUKI algorithm for topology optimization with metaheuristic structural binary distribution. *Eng. Optim.* **2024**, 1–21. [[CrossRef](#)]
13. Shen, L.; Chen, H.; Yu, Z.; Kang, W.; Zhang, B.; Li, H.; Yang, B.; Liu, D. Evolving support vector machines using fruit fly optimization for medical data classification. *Knowl.-Based Syst.* **2016**, *96*, 61–75. [[CrossRef](#)]
14. Aburomman, A.A.; Reaz, M.B. A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems. *Inf. Sci.* **2017**, *414*, 225–246. [[CrossRef](#)]
15. Yang, L.; Zhang, K.; Chen, Z.; Liang, Y. Fault diagnosis of WOA-SVM high voltage circuit breaker based on PCA Principal Component Analysis. *Energy Rep.* **2023**, *9*, 628–634. [[CrossRef](#)]
16. Chauhan, V.K.; Dahiya, K.; Sharma, A. Problem formulations and solvers in linear SVM: A review. *Artif. Intell. Rev.* **2019**, *52*, 803–855. [[CrossRef](#)]
17. Shami, T.M.; El-Saleh, A.A.; Alswaitti, M.; Al-Tashi, Q.; Summakieh, M.A.; Mirjalili, S. Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access* **2022**, *10*, 10031–10061. [[CrossRef](#)]
18. Sun, B.; Li, W.; Huang, Y. Performance of Composite PPSO on Single Objective Bound Constrained Numerical Optimization Problems of CEC 2022. In Proceedings of the 2022 IEEE Congress on Evolutionary Computation (CEC), Padua, Italy, 18–23 July 2022; pp. 1–8.
19. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [[CrossRef](#)] [[PubMed](#)]
20. Speiser, J.L.; Miller, M.E.; Tooze, J.; Ip, E. A comparison of random forest variable selection methods for classification prediction modeling. *Expert Syst. Appl.* **2019**, *134*, 93–101. [[CrossRef](#)] [[PubMed](#)]

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.