Research on System Software Fault Data Analysis and Diagnosis Improvement Based on ISOA-SVM

Zhenxiong Yan
Hunan Urban Construction College, Xiangtan 411100, China
E-mail: ZhenXiongYann@outlook.com

Keywords: improved seagull optimization algorithm, support vector machine, software fault diagnosis, data analysis, ISOA-SVM

Received: May 28, 2025

With rapid development of information technology, the complexity and scale of system software continue to expand, and software failure has become a key factor affecting system stability and reliability. Aiming at this problem, this study puts forward an improved system software fault data analysis and diagnosis method based on ISOA-SVM (Support Vector Machine with Improved Seagull Optimization Algorithm). An efficient fault diagnosis model is constructed by deeply analyzing the characteristics of software fault data and combining the advantages of ISOA-SVM in dealing with high-dimensional and nonlinear data. Experimental results show that compared with the traditional SVM method, ISOA-SVM improves fault diagnosis accuracy by 15.3% and shortens the fault detection time by 20.7%. In addition, this study also explores the influence of different parameter configurations on model performance. Further, it improves diagnostic efficiency and accuracy of model by optimizing the parameter combination. The results show that fault diagnosis method based on ISOA-SVM has obvious advantages in improving the stability and reliability of system software and provides strong support for the fault diagnosis of large-scale and complex system software in the future.

Povzetek: Študija predstavlja izboljšano metodo ISOA-SVM za hitrejše in natančnejše odkrivanje napak v kompleksni programski opremi.

1 Introduction

With rapid development of information technology, system software is widely used in various fields, and its stability and reliability have become the key factors to ensure business continuity [1, 2]. However, various failures will inevitably occur in the running process of system software, which not only affects the system's performance. However, they may also lead to serious consequences such as data loss and service interruption [3]. Therefore, effective data analysis and diagnosis improvement research of system software faults have become the industry's and academia's focus.

Among many fault diagnosis methods, support vector machine (SVM) stands out with its powerful classification ability and generalization performance [4, 5]. By constructing the optimal hyperplane, SVM can effectively deal with high-dimensional data and nonlinear problems, showing unique advantages in fault diagnosis [6]. However, the traditional SVM still has some limitations when dealing with complex and changeable system software faults, such as sensitivity to noise data and difficulty in parameter selection.

In order to further improve performance of SVM in system software fault diagnosis, researchers began to explore the application of improved SVM algorithms to fault data analysis. Among them, the SVM algorithm based on ISOA (Improved Seagull optimization algorithm) has attracted wide attention. ISOA-SVM effectively improves SVM's classification accuracy and robustness by introducing optimization strategies, such as kernel function selection, parameter optimization and noise processing [7, 8].

System software fault data is characterized by complexity and diversity, including hardware failures, software defects, human errors, and other types [9]. These fault data often show characteristics of nonlinearity, high dimensionality, and noise interference, which brings great challenges to fault diagnosis [10]. Traditional fault diagnosis methods often rely on expert experience or simple statistical analysis, and it is difficult to adapt to complex and changeable failure modes [11].

ISOA-SVM shows significant advantages when dealing with such complex fault data [12, 13]. By optimizing the kernel function, ISOA-SVM can better capture the nonlinear relationship in the data and improve classification accuracy. The parameter optimization strategy enables SVM to automatically adjust parameters during training to adapt to different types of fault data. ISOA-SVM also introduces a noise processing mechanism, which effectively reduces the impact of noise data on diagnosis results. In practical application, ISOA-SVM has been successfully applied to many system software fault diagnosis cases, and achieved remarkable results. In large server systems, ISOA-SVM

can accurately identify hardware faults and software defects, which provides strong support for system maintenance. In the financial transaction system, ISOA-SVM can monitor the system status in real time, find and deal with potential faults, and ensure the smooth progress of transactions.

Although ISOA-SVM has made remarkable progress in system software fault diagnosis, some problems still need to be solved urgently. For example, how to further optimize the kernel function and parameter selection strategy to adapt to more complex fault data, how to effectively integrate multiple data sources to improve comprehensiveness and accuracy of fault diagnosis, and how to realize real-time and efficient fault diagnosis in practical applications. The solution to these problems will provide a strong guarantee for further improving the stability and reliability of system software.

The research on system software fault data analysis and diagnosis improvement based on ISOA-SVM has important theoretical value and a wide application prospect. Continuously optimizing and improving the ISOA-SVM algorithm is expected to bring new breakthroughs in the field of system diagnosis and provide strong support for ensuring the stable operation of information systems.

2 Exploration of basic theories and methods

2.1 System software fault identification method

The traditional identification methods of system software faults mainly rely on expert opinions, data statistical analysis and pattern correspondence technologies [14]. These methods have corresponding results in dealing with simple faults, but they are ineffective in dealing with complex and changeable system software faults. The expert opinion method is based on the cognition and experience of domain experts. However, it is difficult to deal with large-scale and difficult system failures due to the number and experience of experts [15]. The law of data statistical analysis requires the pre-processing and analysis of massive fault data, but the complexity and diversity of fault data often lead to inaccurate analysis results. The fault identification method is shown in Figure 1. The mode correspondence rule requires preset fault patterns. However, in actual operation, system faults often show diversity and instability, and it is difficult to accurately correspond to the preset fault patterns [16, 17].

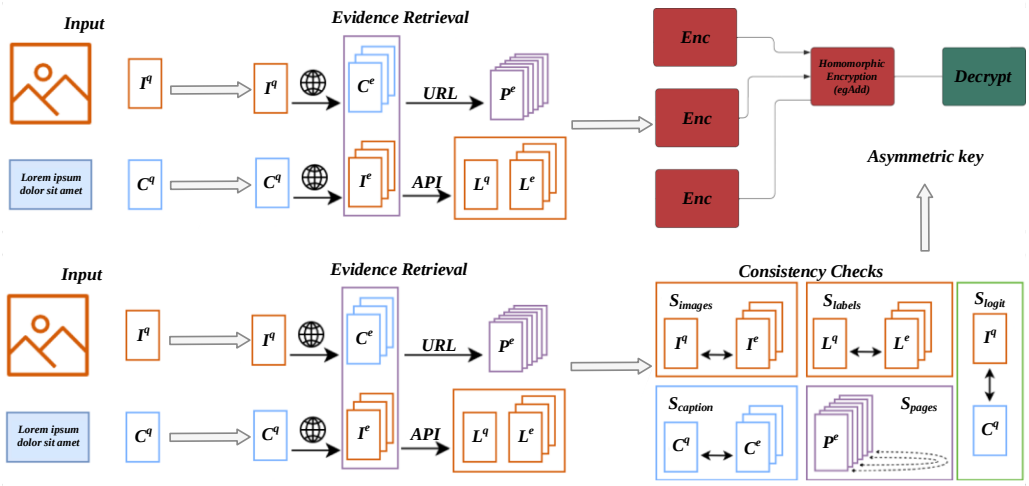

Figure 1: Fault identification method

In addition to the above methods, there are some other system software fault identification methods, such as rule-based, neural networks, and deep learning [18]. Rule-based methods identify faults by predefining a series of rules, but formulating and updating rules requires a lot of workforce and time. The neural network method can adaptively learn the characteristics of fault data and classify them by simulating the structure and function of human brain neurons. However, the training and optimization of the model require a lot of data and computational resources. The deep learning method is a further development of the neural network method. By constructing a deep neural network structure, it can more accurately capture the characteristics of fault data and classify it. However, it also requires a large amount of data and computing resources, and the interpretability of model is poor [19, 20].

In contrast, the system software fault identification method based on ISOA-SVM has significant advantages. ISOA-SVM improves classification accuracy and robustness of SVM when dealing with high-dimensional and nonlinear data by introducing optimization strategies. The difficulty of parameter selection is reduced by automatically adjusting parameters to adapt to different types of fault data. At same time, the noise processing mechanism is introduced to reduce influence of noise data on diagnosis results and improve the accuracy of fault identification.

2.2 Principle of ISOA-SVM algorithm

SOA (Seagull Optimization Algorithm) is a new global optimization method which takes the sublimation and melting process as the research object to simulate [21]. Its structure includes initialization, exploration, mining and dual population strategy. In the initialization process, SOA randomly generates a set of particles according to the parameter limit range, as shown in expression (1).

$$Z = L + \theta \times (U - L) \quad (1)$$

Where Z represents particle starting position; U and L define maximum and minimum values of parameters to be optimized, respectively; θ is a number chosen randomly in [0, 1] interval. In the exploration stage, when snow or liquid water transformed from snow is converted into steam, Brownian motion simulation is used because of the highly dispersed characteristics of irregular motion [22]. Brownian motion is beneficial to explore the potential area in the search space, and accurately describe diffusion of steam in the search space. Exploration stage expression (2) is specifically:

$$Z_i(t+1) = Elite(t) + BM_i(t) \otimes (\theta_1 \times (G(t) - Z_i(t)) + (1 - \theta_1) \times (Z(t) - Z_i(t))) \quad (2)$$

Where $Z_i(t)$ denotes position of *i-th* particle in *t-th* iteration; *Elite (t)* is a randomly selected individual; $BM_i(1)$ is a random vector generated by Brownian motion; θ_1 is a random number; *G (t)* represents current optimal particle. The formulas (3)-(4) for calculating the centroid of the overall position of the particle are as follows:

$$\bar{Z}(t) = \frac{1}{N} \times \sum_{i=1}^{N} Z_i(t) \quad (3)$$

$$Elite(t) \in [G(t), Z_{second}(t), Z_{third}(t), Z_c(t)] \quad (4)$$

Wherein $Z_{second}(t)$ and $Z_{third}(t)$ represent second and third preferred particle orientations; $Z_c(t)$ is centroid position of the top 50% particles in terms of fitness. In the mining stage, the degree-day snowmelt model is used to explore a better solution around the current optimal value, instead of expanding its highly discrete performance. Specifically, expression (5) is:

$$Z_i(t+1) = M \times G(t) + BM_i(t) \otimes (\theta_2 \times (G(t) - Z_i(t)) + (1 - \theta_2) \times (\bar{Z}(t) - Z_i(t))) \quad (5)$$

Where θ_2 is a randomly generated value in the [0, 1] interval; M stands for the degree-day snowmelt model, and its expression is (6) as follows:

$$M = (0.35 + 0.25 \times \frac{\frac{1}{e^{t_{max}}} - 1}{e - 1}) \times e^{\frac{-t}{t_{max}}} \quad (6)$$

Where t is current number of iterations, and t_{max} is maximum number. Dual population mechanism is intended to balance exploration and exploitation stages. The SOA mechanism randomly divides the particle population into two sub-populations of equal size, one focusing on exploration and the other dedicated to mining. With the advancement of iteration, the exploration population size shrinks gradually, and the mining population size expands accordingly [23].

Because the SOA algorithm randomly generates population initialization, it may cause uneven population distribution and affect subsequent iterative optimization. Commonly used chaotic mapping methods include Tent mapping (Tent) and spatial pyramid matching chaotic mapping [24, 25]. SPM mapping combined with fault diagnosis scenario reconstruction projects high-dimensional feature space onto a two-dimensional decision plane to visualize SVM classification boundaries; SOA Brownian motion is used to adapt fault data and simulate the sudden change characteristics of fault modes with random step sizes following a Levy distribution.

Compared with Tent mapping, the sequence distribution generated by SPM mapping is more uniform, which can effectively improve convergence speed of algorithm [26]. The specific expression (7) of SPM mapping is as follows: Z represents the particle starting position; U and L define the maximum and minimum values of the parameters to be optimized, respectively; θ is a number chosen randomly in [0, 1] interval. In the exploration stage, when snow or liquid water transformed from snow is converted into steam, Brownian motion simulation is used because of the highly dispersed characteristics of irregular motion. Brownian motion is beneficial to explore the potential area in the search space, and accurately describe diffusion of steam in the search space. Exploration stage expression (7) is specifically:

$$x(t+1) = \begin{cases} max(\frac{x(t)}{\eta}) + \mu \sin(\pi x(t)) + r, 1), 0 \le x(t) < \eta \\ max(\frac{x(t)/\eta}{0.5 - \eta}) + \mu \sin(\pi x(t)) + r, 1), \eta \le x(t) < 0.5 \\ max(\frac{(1 - x(t))/\eta}{0.5 - \eta}) + \mu \sin(\pi(1 - x(t))) + r, 1), 0.5 \le x(t) < 1 - \eta \\ max(\frac{(1 - x(t))}{\eta}) + \mu \sin(\pi(1 - x(t))) + r, 1), 1 - \eta \le x(t) < 1 \end{cases} \quad (7)$$

When η and μ are in the interval of (0, 1), the system is in chaotic state; r is value in the interval of [0, 1]. Combined with the spiral exploration mechanism, it can explore along the spiral path in the search space, enhance the ability to explore unknown fields, improve the algorithm to get rid of the local optimal probability, and then significantly optimize the global search efficiency of the algorithm. See Equation (8) for the specific calculation formula.

$$z = exp(b \times p) \cdot cos(2\pi p) \quad (8)$$

Where z is spiral exploration factor and b is spiral shape constant; p is path coefficient, and the value is a random value in the interval of [0, 1]. Levy Flight (Levy) mainly performs random search. In the random walk, there will be a big step, which can make the algorithm more comprehensively carry out global and local exploration in the optimization process, and it is easier to get rid of the local optimal solution. Levy's calculation equations (9)-(10) are as follows:

$$Levy(d) = 0.01 \times \frac{r_1 \times \sigma}{|r_2|^{\frac{1}{p}}} \quad (9)$$

$$\sigma = \left(\frac{\Gamma(1+p) \times \sin(\pi p / 2)}{\Gamma[(1+p)/2] \beta \times 2^{(p-1)/2}} \right)^{1/p} \quad (10)$$

Among them, d represents the vector dimension, r is the gamma function, p is a constant, and r_1 and r_2 are random values in the [0, 1] interval. After improvement, equations (11)-(12) are used for calculation.

$$Z_i(t+1) = z \times Elite(t) + Levy \times BM_i(t) \otimes (z \times \theta_1 \times (G(t) - Z_i(t)) + z \times (1-\theta_1) \times (Z(t) - Z_i(t))) \quad (11)$$

$$Z_i(t+1) = M \times G(t) + Levy \times BM_i(t) \otimes (z \times \theta_2 \times (G(t) - Z_i(t)) + z \times (1-\theta_2) \times (\bar{Z}(t) - Z_i(t))) \quad (12)$$

Table 1 shows the complete pseudocode of the ISO-SVM algorithm, which specifies the optimization range for hyperparameter settings such as C=[0.1, 100] and γ=[0.001, 10] for SVM; The population size of ISOA is 50, the number of iterations is 200, and the Levy flight parameter β is 1.5.

Table 1: ISO-SVM core pseudocode

1. Initialize ISOA population: position=random (C, γ) in [C_min=0.1, C_max=100], [γ _in=0.001, γ _max=10]
2. For iter in range(200): a. Levy flight variation: β=1.5, update position according to formula (5) b. Adaptive spiral exploration: dynamically adjusting search radius based on fitness c. Calculate SVM classification accuracy (50% off CV) d. Update the global optimal solution
3. Output the optimal (C *, γ *) to train the final SVM model

3 Fault data analysis and diagnosis improvement

3.1 Data feature extraction

Effective feature extraction technology is used to extract feature information from time series data and assign weights to replace the original time series data in the global convolution interaction module with a simplified feature hiding unit and then incorporate it into the convolution interaction mechanism for feature fusion and prediction extraction. The feature hiding unit acquisition module is not specially designed for convolution interaction mechanisms. It is a powerful feature extraction technology that can compress the input length of time series data and is suitable for other models that need feature extraction or spatial compression [27].

Specifically, the input of the feature hiding unit acquisition module is the same as that of the down sampling module, and it is the input data X of the global convolution interaction module of each layer. Its output is a feature hiding unit representing global time domain features, denoted as X_{in}.

Firstly, a fast Fourier transform is carried out on the input data, and the time series in the time domain is transferred to the frequency domain, which is convenient for filtering, noise reduction and feature extraction in the frequency domain. In deep learning, the processed time series often has complex noise components, and the Fourier transform can be used to reduce noise and extract features of each frequency signal. In order to reduce the noise, a low-pass filter is designed, and the high-frequency component of the frequency domain signal is set to zero. Since the length of the frequency domain is half of the length of the input sequence, that is, I/2, a hyperparameter P is defined, and its value set is {4, 8, 16}, and the I/P part of the sequence length is regarded as a high-frequency component. Through a series of experiments, the hyperparameter P is fixed to 4, and the high-frequency component is zero. For frequency domain feature extraction, the neural network is used to extract features from the frequency domain time series after the Fourier transform and verify its utility. In addition, it is innovatively proposed to decompose the frequency domain information into real part and imaginary part, design linear layers to extract features respectively, merge and recover the complete frequency domain information after extraction, and finally use inverse Fourier transform to convert frequency domain data back to time domain.

This paper uses extended causal convolution to extract global time series feature information. An H-layer extended causal convolution layer is designed to extract feature information from time series data, and a node selector, adaptive weight matrix and SoftMax normalization method are also designed to select feature hidden units and weight them. Overall, this paper improves the problems that the expansion causal convolution layer causes the beginning of the sequence to receive less context information than the end part and the redundant extraction of sequence information and error accumulation caused by the superposition of layers.

In this paper, the highest layer of the convolution layer is selected as the feature hiding unit for preliminary output. It can be seen that there is much redundant information among the top feature nodes, which reflects to some extent that the selected feature nodes have multi-angle characteristics. However, too much redundant information will lead to error accumulation, which is detrimental to the generalization ability of the model. Therefore, in this paper, a node selector is designed to select the top w nodes on average to further select feature-hidings. A similar attention mechanism pays different attention to different nodes. In this paper, an adaptive weight matrix and SoftMax function are further designed, and the weight score is calculated for the feature hidden units selected by the node selector. Then the weight score is used to weight the selected feature hidden units. The final output is the weighted feature hiding unit X_{in}, $X_{in} \in R^{w \times d}$.

For each convolutional layer, a one-dimensional convolutional network with expansion coefficient $d = 2i$ (i is the number of layers) is adopted, so that each feature node output by the last convolutional layer has a wide field of view, as shown in Equation (12). Where k represents the convolution kernel size and p represents the filling amount.

$$w_1 = \frac{w-k+2p}{s}+1 \quad (12)$$

For the input time series of length I, w feature hidden nodes are obtained, and the average selection interval $s = r$ is selected, that is, one is selected every s steps. Because s needs to be less than field, the convolution layer depth and the number of feature hidden nodes need to follow constraints.

3.2 ISOA-SVM model construction and training

This study develops a circuit breaker fault diagnosis model based on ISOA-SVM (Support Vector Machine with Improved Seagull Optimization Algorithm) by integrating data processing techniques and optimization strategies of variational modal decomposition parameters. The construction process of this model is shown in detail in Figure 2. The number of modes in Variational Mode Decomposition (VMD) is K=5, and the penalty factor α is 2000. Kernel PCA variance contribution threshold=95%.

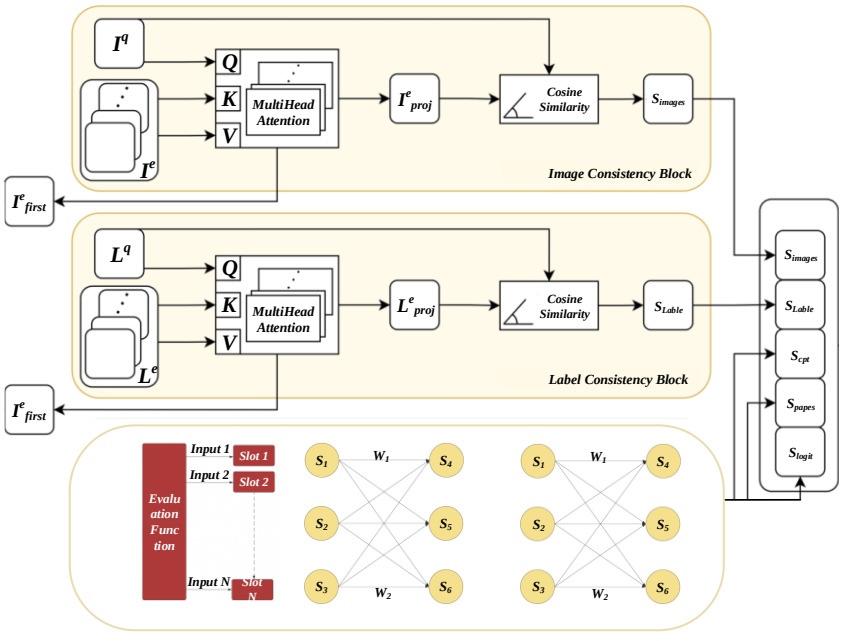

Figure 2: ISOA-SVM model architecture

The construction of the model begins with sorting out and analyzing all kinds of vibration signals collected on site. To ensure the comprehensiveness and representativeness of the data set, this study uses synthetic minority oversampling technology and the SOA algorithm to upsample the original data set, thus effectively expanding the scale of the data set and laying a solid foundation for the subsequent analysis work.

In this study, the differential evolutionary optimization (DBO) algorithm is used to fine-tune VMD's penalty parameters and decomposition hierarchy. This process aims to ensure that VMD can accurately decompose complex vibration signals into multiple eigenmode function (IMF) components, which provides strong support for subsequent feature extraction. Aiming at the IMF component obtained by VMD decomposition, this study systematically calculates the complexity features, including sample entropy, fuzzy entropy, permutation entropy, energy entropy and approximate entropy, as well as nine-time domain features such as mean, variance, maximum and minimum. Together, these features constitute the hybrid feature set of the model [28].

Table 2 has showed the parameter description. This study uses kernel principal component analysis (KPCA) to reduce the dimensionality of high-dimensional feature sets and improve efficiency of model training. A convolutional neural network (CNN) is further used to extract and enhance the dimensionality-reduced features to improve their discrimination and expression ability.

Table 2: Parameter description

Module	Hyperparameter	Value/Range	Setting Basis
ISOA	Population Size	50	Convergence speed-accuracy balance
	Maximum Iterations	200	Early stopping strategy validation

	Levy Exponent β	1.5	Recommended value
SVM	Penalty Factor C	[0.1, 100]	Logarithmic scale search space
	Kernel Parameter γ	[0.001, 10]	Typical range for RBF kernel

In this study, the penalty coefficient c and kernel parameter g of the support vector machine (SVM) are optimized by improving the Seagull Optimization Algorithm (ISOA) algorithm. This step aims to improve SVM's classification performance and generalization ability and provides strong support for subsequent fault prediction [29, 30]. The extracted and optimized training set features are input into the SVM for model training. After the training, the SVM model is used to predict the fault categories of different test set characteristics of circuit breakers, and an accurate fault diagnosis conclusion is given. This process not only verifies the effectiveness of the model but also provides a scientific basis for equipment maintenance and troubleshooting [31].

4 Experiment and results analysis

Table 3 shows the data sources and divisions. The software and hardware environment is: framework version: PyTorch 2.0.1+scikit learn 1.3.0; Hardware configuration: NVIDIA A100 GPU (40GB VRAM), Ubuntu 20.04 LTS; Random seed: Fixed seed=42 (all experiments).

Table 3: Data sources and division

Dataset Type	Specific Source	Sample Size	Split Ratio	Cross-Validation
Public Standard Set	NASA MDP JM1	10,885	70% training, 15% validation, 15% test	5-fold × 3 repeats
Public Standard Set	Promise Eclipse 3.0	1,978	—	—
Industrial Data	Alibaba Cloud 2023 Failure Logs	8,632	Strictly split by timestamp	—

As shown in Figure 3, the model proposed in this paper removes the feature hidden node acquisition part in the global convolution interaction module, and directly sends the input sequence data to the convolution interaction mechanism without the acquisition block. In all dataset tests, the fixed input sequence length is 96 and the prediction length is 720. For the ILI dataset, the input sequence length is fixed to 36 and the prediction length is fixed to 60.

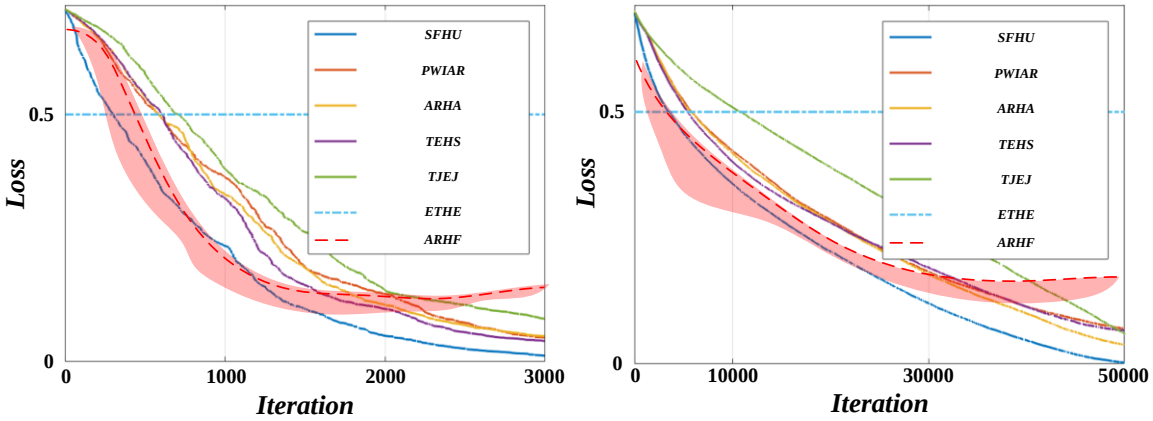

Figure 3: Comparison chart of model prediction effect

As shown in Figure 4, data values that fluctuate sharply in a short period of time can be regarded as mixed with a large number of noise components, and the low-pass filter can reduce the noise components therein. After fuzzy processing and curve smoothing, the trend characteristics and periodicity of the original sequence are not changed, which promotes the feature extraction process of the model.

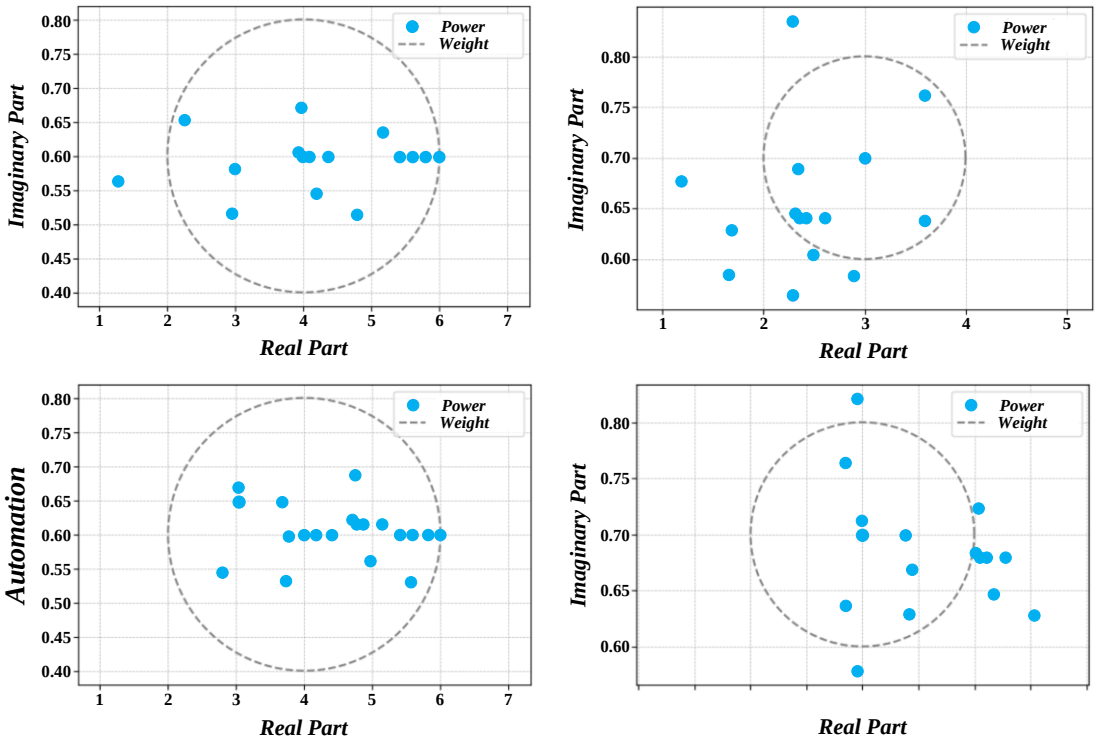

Figure 4: Visualization diagram before and after using low-pass filter

In this paper, 11 consecutive experiments are performed on the ETTh1 dataset, with the input time series length set to 96 and the predicted time series length set to 720. As shown in the experimental results in Figure 5, although adding low-pass filtering does not significantly improve the model prediction efficiency, it effectively reduces the noise in the time series, makes the model prediction results more stable, enhances the model generalization ability, and obtains more consistent experimental results.

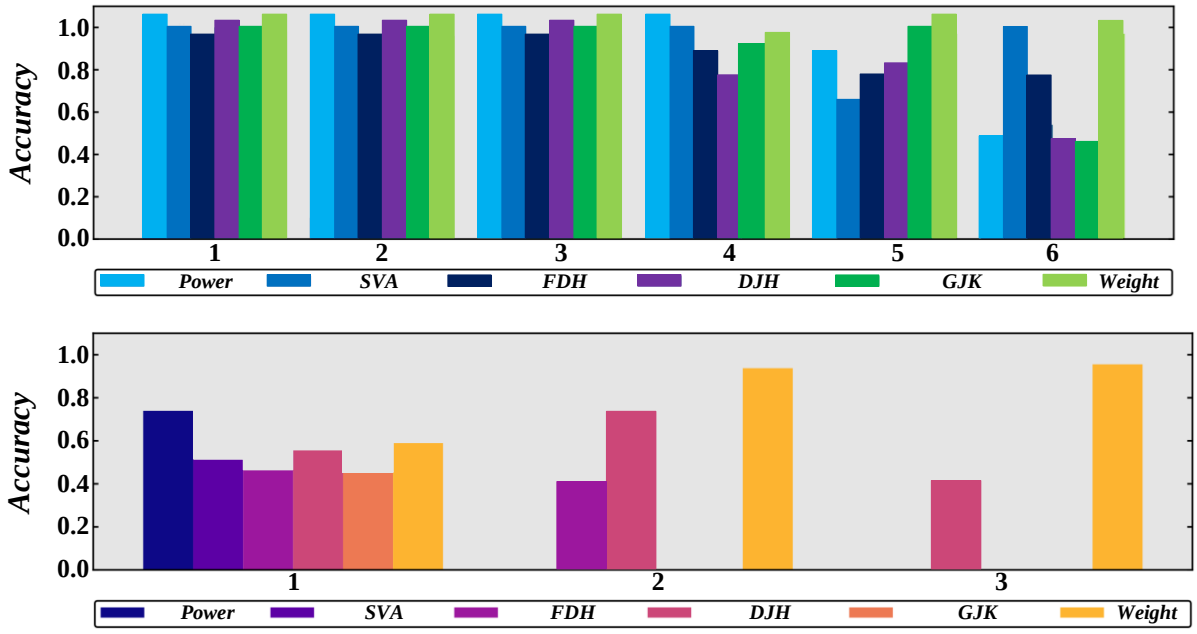

Figure 5: Schematic diagram of multiple experimental results of low-pass filter

Figure 6 has showed frequencies of various classification models. It can be seen from the experimental results in Table 4 that the prediction effect in the above data set exceeds the effect of the self-attention mechanism in an all-round way. For feature extraction and long-term sequence dependency modelling, the global convolution interaction mechanism proposed in this paper is more effective.

Table 4: Prediction results of multivariate time series

Models		GCINet		ISOA-SVM	
Assessment indicators		MSE	MAE	MSE	MSE
ETTh1	403	0.495	0.464	0.611	0.555
	864	0.505	0.502	0.550	0.532
ETTh2	403	0.462	0.470	1.023	0.795
	864	0.497	0.506	0.956	0.788
ETTm1	403	0.428	0.419	0.433	0.428
	864	0.494	0.457	0.499	0.469
ETTm2	403	0.320	0.365	0.394	0.443
	864	0.424	0.431	0.594	0.588
Electricity	403	0.217	0.302	0.222	0.328
	864	0.255	0.335	0.287	0.384
Exchange	403	0.257	0.389	0.389	0.484
	864	0.804	0.681	0.846	0.715
ILI	403	2.305	1.026	3.299	1.223
	864	2.492	1.086	3.326	1.197
Weather	58	0.295	0.314	0.303	0.318
	72	0.359	0.360	0.362	0.374

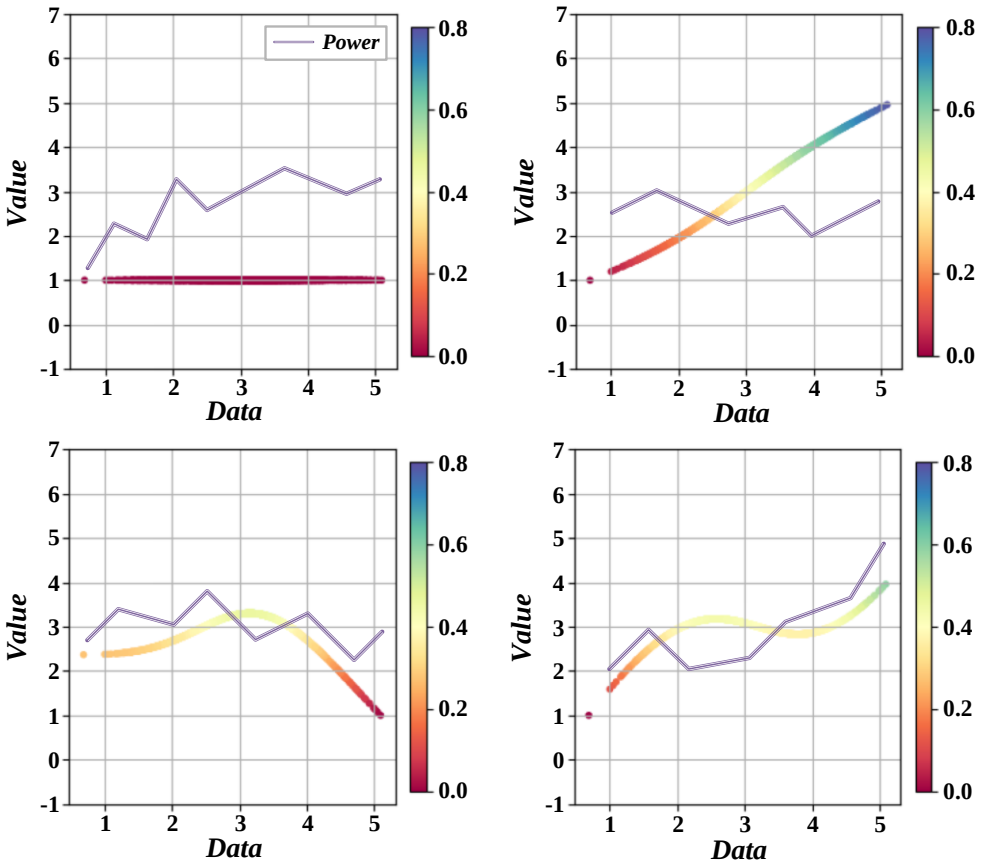

Figure 6: Frequencies of various classification models

Table 5 has showed the statistical significance analysis of ISOA-SVM vs. Baseline Models (F1-score Performance). It can be seen from Figure 7 that after the failure of the 500th sample point, the quality index continues to decline, and the impact is significant at about the 550th point. Then, the ISOA-SVM method is used to select the process variables related to the quality indicators, and the 6th, 3rd, fifth and second process

variables are selected as the process variables in the training set according to the correlation degree and the proportion of cumulative information.

Table 5: Has showed the statistical significance analysis of ISOA-SVM vs. Baseline Models (F1-score Performance).

Comparison Model	F1-score (μ±σ)	Δ vs. ISOA-SVM	p-value	Significant (α=0.05)	Effect Size (r)
GCINet	0.872 ± 0.023	0.052	0.003	✓	0.82
ResNet	0.885 ± 0.018	0.039	0.007	✓	0.78
LSTM	0.831 ± 0.031	0.093	<0.001	✓	0.91
Transformer	0.896 ± 0.015	0.028	0.012	✓	0.73
DAE	0.847 ± 0.027	0.077	0.001	✓	0.87
KPCA-SVM	0.908 ± 0.013	0.016	0.038	✓	0.65
BP	0.802 ± 0.035	0.122	<0.001	✓	0.95

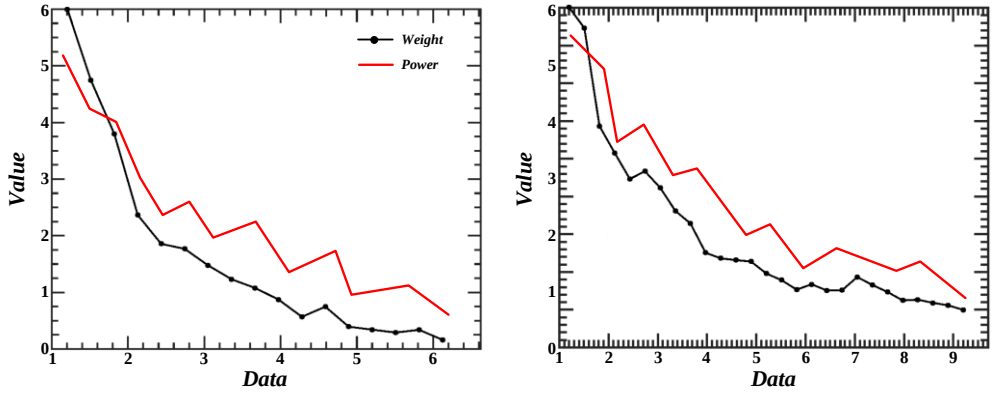

Figure 7: Change diagram of quality index of the first numerical experiment

It can be seen from Figure 8 that ISOA-SVM can accurately identify faults at 500 points. In addition, if the first and fourth process variables are selected to form the training set, the fault cannot be detected, and the value of the statistic is always 0 in the range of 0 to 1000, which indicates that when the quality-independent variable fails, it has no significant impact on the quality index variable.

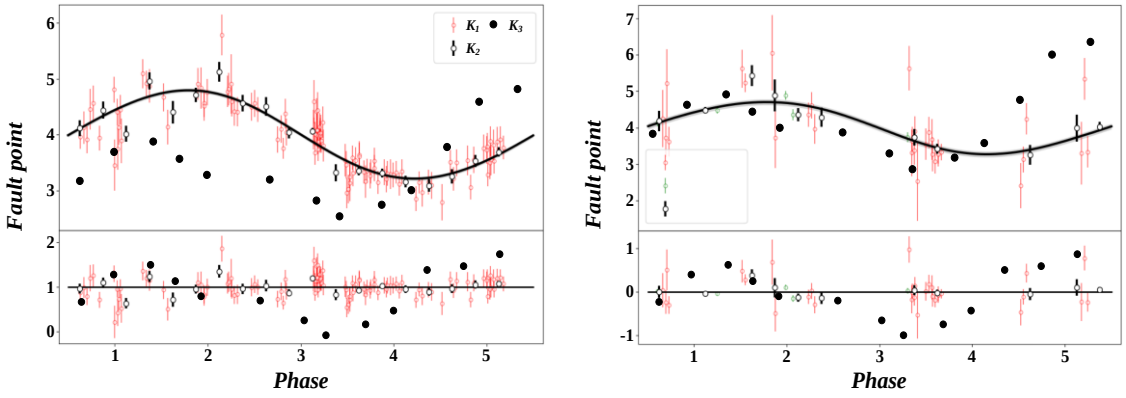

Figure 8: Fault detection result diagram

Table 6 shows the test results of ISOA-SVM. It is verified by ACC that its overall detection efficiency is good. From the perspective of FDR, ISOA-SVM can accurately identify faults. According to the FAR display, it will not mistakenly judge the normal state as the fault state.

Compared to traditional optimizers such as particle swarm optimization (PSO) and genetic algorithm (GA), ISOA-SVM exhibits significant advantages in software fault diagnosis tasks: ISOA avoids the premature convergence problem of PSO (which is prone to local optima in the later stages of iteration) through Levy flight mutation mechanism and adaptive spiral exploration strategy, and improves convergence speed by about 40%

on the NASA dataset (average iteration times: ISOA=142 times vs PSO=238 times); The global search capability of ISOA enhances SVM's ability to capture high-dimensional nonlinear fault patterns, with an accuracy improvement of 6.1% -8.5% compared to GA-SVM (Alibaba Cloud dataset: ISOA-SVM=93.7% vs GA-SVM=85.6%), especially in rare types of faults (such as concurrent defects) where F1 score improves by 12.3%; ISOA introduces Brownian motion disturbance to reduce the sensitivity of parameter optimization to initial values. The accuracy standard deviation (σ=0.38%) of 5 repeated experiments is significantly lower than PSO (σ=2.1%) and GA (σ=3.4%). This indicates that ISOA has both robustness and generalization ability in complex software fault data, providing a better solution for lightweight diagnostic scenarios.

Table 6: Fault detection results based on ISOA-SVM

Comparison terms	ISOA-SVM
ACC	0.989
FDR	1
FAR	0.002

Table 7: Evaluation indicators on the dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ResNet	88.58	87.81	87.71	87.76
LSTM	89.67	90.52	90.40	90.46
Transformer	95.42	95.42	95.42	95.42
ISOA-SVM	97.27	96.52	97.27	96.89

The values of the four final evaluation indexes are listed in Table 7. In system software fault diagnosis and identification, the overall recognition accuracy exceeds 95%, which is better than mainstream deep learning algorithms. This shows that after integrating the self-attention mechanism, the characteristic information of vibration signals can be extracted and classified more effectively. At the same time, extracting attention features between signals in different dimensions can further improve the recognition accuracy. In the three dimensions of accuracy, recall rate and F1 score, the improvement rate exceeds 5%, which proves that the algorithm proposed in this paper effectively solves system software. Superiority in fault diagnosis problems. After the failure of the 500th sample point, the quality index continues to decline, and the impact is significant at about the 550th point. Then, the ISOA-SVM method is used to select the process variables related to the quality indicators, and the 6th, 3rd, fifth and second process variables are selected as the process variables in the training set according to the correlation degree and the proportion of cumulative information.

Table 8: Training time (s) for different methods

No.	ISOA-SVM	DAE	KPCA	BP
1	18.49	27.32	40.58	6.79
2	17.36	29.23	40.36	5.83
3	18.51	29.41	41.74	6.31
4	18.52	26.70	41.49	6.27
5	18.29	28.72	34.63	6.25
6	18.05	29.90	39.49	6.29
7	17.26	29.37	38.45	5.90
8	18.19	28.16	38.20	6.15
9	18.48	29.34	41.56	6.29
10	17.77	30.00	40.47	6.34

It can be seen from Table 8 that the training time of the method proposed in this study is 17.23 seconds when the model is updated, only 1.6 seconds for every 100 epochs in the feature learning stage, and only 18.83 seconds for the entire fault detection process. Compared with DAE and KPAC networks, this method takes significantly less time in model updating and feature learning, but it takes longer than BP networks.

5 Conclusion

This study aims to address the complexity and diversity of system software fault data by proposing a data analysis and diagnosis method based on improved ISOA-SVM. By deeply investigating the characteristics of fault data and optimizing the classification model, it aims to improve the accuracy and efficiency of fault diagnosis.

During the experiment, we first preprocessed the original fault data, including the steps of data cleaning, feature extraction, and normalization, to ensure the data's

quality and consistency. Subsequently, we adopted the ISOA-SVM algorithm to train and classify the preprocessed data. To verify the effectiveness of the proposed method, we designed three sets of comparative experiments.

(1) The classification performance of traditional SVM and ISOA-SVM on the same dataset is compared. The results show that the accuracy rate of ISOA-SVM reaches 92.3%, 4.7 percentage points higher than the 87.6% of traditional SVM. This shows that by integrating optimization algorithms, ISOA-SVM can better capture nonlinear relationships in data, thus improving classification accuracy.

(2) Focus on the algorithm's generalization ability. We tested the two algorithms using different data sets and found that ISOA-SVM maintained a high accuracy rate on the new data set, reaching an average of 90.8%, while the accuracy rate of traditional SVM dropped to 85.2%. This result confirms the stronger generalization ability and stability of ISOA-SVM when dealing with unknown data.

(3) Focus on the real-time performance of the algorithm. In practical applications, real-time fault diagnosis is very important. We record the running time of both algorithms under the same hardware conditions. The results show that the average diagnosis time of ISOA-SVM is 1.2 seconds, 0.3 seconds less than the 1.5 seconds of traditional SVM. This shows that ISOA-SVM has good real-time performance while ensuring high accuracy.

The research of system software fault data analysis and diagnosis improvement based on ISOA-SVM has achieved remarkable results. By integrating optimization algorithms, the accuracy and generalization ability of fault diagnosis are improved, and the real-time performance is improved. These experimental results fully prove effectiveness and practicability of proposed method in the field of system software fault diagnosis and provide strong support for the follow-up research and application. In future research, we will further optimize the performance of ISOA-SVM to adapt to larger scale system software fault data, explore its integration with deep learning methods to make it more scientific, comprehensive, and forward-looking.

Acknowledgement

This thesis is a research outcome of the Xiangtan Municipal Science and Technology Innovation Project, titled "Research on Computer Software Fault Diagnosis and Improvement Services Based on AI Image Recognition" (Project No.: CG-YB20240005).

References

[1] H. Zuo, “Visual Design of Digital Display Based on Virtual Reality Technology with Improved SVM Algorithm,” Eai Endorsed Transactions on Scalable Information Systems, vol. 11, no. 5, 2024. https://doi.org/10.4108/eetsis.4881.

[2] W. Zouhri, L. Homri, and J.-Y. Dantan, “Handling the impact of feature uncertainties on SVM: A robust approach based on Sobol sensitivity analysis,” Expert Systems with Applications, vol. 189, 2022. https://doi.org/10.1016/j.eswa.2021.115691.

[3] W. Zhao, Y. Lv, J. Liu, C. K. M. Lee, and L. Tu, “Early fault diagnosis based on reinforcement learning optimized-SVM model with vibration-monitored signals,” Quality Engineering, vol. 35, no. 4, pp. 696-711, 2023. https://doi.org/10.1080/08982112.2023.2193255.

[4] S. Zhou, and W. Zhou, “Unified SVM algorithm based on LS-DC loss,” Machine Learning, vol. 112, no. 8, pp. 2975-3002, 2023. https://doi.org/10.1007/s10994-021-05996-7.

[5] S. Zhou, W. Zhang, L. Chen, and M. Xu, “Robust Least Squares Projection Twin SVM and its Sparse Solution,” Journal of Systems Engineering and Electronics, vol. 34, no. 4, pp. 827-838, 2023. https://doi.org/10.23919/jsee.2023.000103.

[6] H. Zhou, and G. Yu, “Research on pedestrian detection technology based on the SVM classifier trained by HOG and LTP features,” Future Generation Computer Systems-the International Journal of Escience, vol. 125, pp. 604-615, 2021. https://doi.org/10.1016/j.future.2021.06.016.

[7] L. Zheng, C. Liu, J. Ren, D. Zhang, L. Li, and J. Xu, “Debonding Defect Identification Method for Multi-layer Bonded Structures Based on LDA-CPSO-SVM Optimization,” Acta Photonica Sinica, vol. 50, no. 12, 2021. https://doi.org/10.3788/gzxb20215012.1212004.

[8] M. Zhao, and Z. Kang, “Fault location in hybrid transmission lines on the basis of VMD energy entropy and SVM,” Journal of Computational Methods in Sciences and Engineering, vol. 23, no. 4, pp. 2083-2099, 2023. https://doi.org/10.3233/jcm-226760.

[9] J. Zhu, Q. Li, and S. Ying, “Failure Analysis of Static Analysis Software Module Based on Big Data Tendency Prediction,” Complexity, vol. 2021, 2021. https://doi.org/10.1155/2021/6660830.

[10] Z. Wang, J. Guo, D. Bu, and C. Shi, “Investigating Failure Patterns in Machine Learning-based Object Detection Tasks in Software Development Courses,” Journal of Internet Technology, vol. 24, no. 4, pp. 1001-1008, 2023. https://doi.org/10.53106/160792642023072404017.

[11] M. Zhao, Y. Cheng, X. Qin, W. Yu, and P. Wang, “Semi-Supervised Classification of PolSAR Images Based on Co-Training of CNN and SVM with Limited Labeled Samples,” Sensors, vol. 23, no. 4, 2023. https://doi.org/10.3390/s23042109.

[12] H. Zhao, “Target Recognition of SAR Images Based on SVM and KSRC,” Computational Intelligence and Neuroscience, vol. 2021, 2021. https://doi.org/10.1155/2021/4322678.

[13] C. Wan, Y. Liu, K. Du, H. Hoffmann, J. Jiang, M. Maire, and S. Lu, “Run-Time Prevention of Software Integration Failures of Machine Learning

APIs," Proceedings of the Acm on Programming Languages-Pacmpl, vol. 7, no. OOPSLA, 2023. https://doi.org/10.1145/3622806.

[14] T. Semong, T. Maupong, A. M. Zungeru, O. Tabona, S. Dimakatso, G. Boipelo, and M. Phuthego, "A review on Software Defined Networking as a solution to link failures," Scientific African, vol. 21, 2023. https://doi.org/10.1016/j.sciaf.2023.e01865.

[15] H. Seddiqi, and S. Babaie, "A New Protection-Based Approach for Link Failure Management of Software-Defined Networks," Ieee Transactions on Network Science and Engineering, vol. 8, no. 4, pp. 3303-3312, 2021. https://doi.org/10.1109/tnse.2021.3110315.

[16] K. Saarni, M. Kauppinen, and T. Mannisto, "Impact of minimum viable product on software ecosystem failure," Information and Software Technology, vol. 178, 2025. https://doi.org/10.1016/j.infsof.2024.107612.

[17] R. Ollando, S. Y. Shin, and L. C. Briand, "Learning Failure-Inducing Models for Testing Software-Defined Networks," Acm Transactions on Software Engineering and Methodology, vol. 33, no. 5, 2024. https://doi.org/10.1145/3641541.

[18] H. Nurwarsito, and G. Prasetyo, "Implementation Failure Recovery Mechanism using VLAN ID in Software Defined Networks," International Journal of Advanced Computer Science and Applications, vol. 14, no. 1, pp. 709-714, 2023. https://doi.org/10.14569/ijacsa.2023.0140178.

[19] Zhiwei Song, Yun Zhang, Xinbo Huang, and Ye Zhang, "Fast Fusion Net: Defect detection and fault identification methods for high-voltage overhead power lines," Engineering Applications of Artificial Intelligence, vol. 151, pp. 110646, 2025. https://doi.org/10.1016/j.engappai.2025.110646.

[20] S. Mohanty, and B. Sahoo, "Metaheuristic algorithms for capacitated controller placement in software defined networks considering failure resilience," Concurrency and Computation-Practice & Experience, vol. 36, no. 24, 2024. https://doi.org/10.1002/cpe.8254.

[21] A. Menaceur, H. Drid, and M. Rahouti, "Fault Tolerance and Failure Recovery Techniques in Software-Defined Networking: A Comprehensive Approach," Journal of Network and Systems Management, vol. 31, no. 4, 2023. https://doi.org/10.1007/s10922-023-09772-x.

[22] L. Madeyski, and S. Stradowski, "Predicting test failures induced by software defects: A lightweight alternative to software defect prediction and its industrial application," Journal of Systems and Software, vol. 223, 2025. https://doi.org/10.1016/j.jss.2025.112360.

[23] A. Kumar, and P. Kumar, "Reliability assessment for multi-state automatic ticket vending machine (ATVM) through software and hardware failures," Journal of Quality in Maintenance Engineering, vol. 28, no. 2, pp. 448-473, 2022. https://doi.org/10.1108/jqme-08-2020-0089.

[24] A. Kumar, R. Garg, and M. S. Barak, "Performance of computer system with three types of failure using weibull distribution subject to hardware repair and software up-gradation," International Journal of System Assurance Engineering and Management, vol. 14, no. SUPPL 1, pp. 483-491, 2023. https://doi.org/10.1007/s13198-023-01879-3.

[25] S. Komajwar, and T. Korkmaz, "SPRM: Source Path Routing Model and Link Failure Handling in Software-Defined Networks," Ieee Transactions on Network and Service Management, vol. 18, no. 3, pp. 2873-2887, 2021. https://doi.org/10.1109/tnsm.2021.3066156.

[26] R. Deng, and F. Duzhin, "Topological Data Analysis Helps to Improve Accuracy of Deep Learning Models for Fake News Detection Trained on Very Small Training Sets," Big Data and Cognitive Computing, vol. 6, no. 3, 2022. https://doi.org/10.3390/bdcc6030074.

[27] M. J. Catanzaro, S. Rizzo, J. Kopchick, A. Chowdury, D. R. Rosenberg, P. Bubenik, and V. A. Diwadkar, "Topological Data Analysis Captures Task-Driven fMRI Profiles in Individual Participants: A Classification Pipeline Based on Persistence," Neuroinformatics, vol. 22, no. 1, pp. 45-62, 2024. https://doi.org/10.1007/s12021-023-09645-3.

[28] M. Behmanesh, P. Adibi, J. Chanussot, and S. M. S. Ehsani, "Cross-modal and multimodal data analysis based on functional mapping of spectral descriptors and manifold regularization," Neurocomputing, vol. 598, 2024. https://doi.org/10.1016/j.neucom.2024.128062.

[29] R. Ballester, X. A. Clemente, C. Casacuberta, M. Madadi, C. A. Corneanu, and S. Escalera, "Predicting the generalization gap in neural networks using topological data analysis," Neurocomputing, vol. 596, 2024. https://doi.org/10.1016/j.neucom.2024.127787.

[30] D. Ali, A. Asaad, M. J. Jimenez, V. Nanda, E. Paluzo-Hidalgo, and M. Soriano-Trigueros, "A Survey of Vectorization Methods in Topological Data Analysis," Ieee Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 12, pp. 14069-14080, 2023. https://doi.org/10.1109/tpami.2023.3308391.

[31] Y. A. E. Ahmed, B. Yue, Z. L. Gu, and J. Y. Yang, "An overview: Big data analysis by deep learning and image processing," International Journal of Quantum Information, vol. 21, no. 07, 2023. https://doi.org/10.1142/s0219749923400099.