https://doi.org/10.31449/inf.v50i11.9105 Informatica **50** (2026) 53–72 **53**

Automobile Fault Diagnosis Using Lightweight Cetacean Optimization and Multi-Scale Residual Neural Networks

Zhe Chen
Automobile School, Zhejiang Institute of Communications, Hangzhou 311112, China
E-mail: ZheeChenn@outlook.com

Keywords: Automobile fault diagnosis, lightweight cetacean algorithm, intelligent algorithms

Received: May 2, 2025

With the advancement of automotive technologies, accurate and real-time fault diagnosis is essential for ensuring vehicle safety and reducing maintenance costs. This paper proposes an automobile fault diagnosis system based on the Lightweight Whale Optimization Algorithm (LWA) integrated with a Multi-Scale Residual Unit deep neural network. The proposed model is trained on a real-world automobile sensor dataset containing 2,000 labeled samples spanning four fault types: engine, brake system, battery, and transmission failures. The model demonstrates a diagnostic accuracy of 95.4%, outperforming baseline methods such as Support Vector Machine (90.1%), Random Forest (92.3%), and CNN-based models (94.2%). Additionally, the LWA achieves faster convergence and a 25% reduction in inference time compared to traditional MSRU models, with a response time under 2.5 seconds. The lightweight design also reduces model parameters by 64%, enabling real-time deployment in embedded vehicle systems. Experimental results show that the proposed method not only enhances diagnostic accuracy but also improves computational efficiency, offering a practical solution for intelligent automotive maintenance.

Povzetek: Prispevek predstavi učinkovit in lahek model za diagnostiko okvar vozil, ki z visoko natančnostjo in hitrostjo izboljšuje zanesljivost ter omogoča uporabo v realnem času.

1 Introduction

With the rapid development of the automobile industry as an essential means of transportation in modern society, automobiles' safety, reliability, and service life have become increasingly important. Traditional automobile fault diagnosis methods, such as rule-based models, typically achieve a diagnostic accuracy of around 85% for common faults. However, when faced with complex and diversified faults, these methods often experience lower accuracy and slower response times, which hinder their effectiveness in real-time applications. This inadequacy highlights the need for more accurate and efficient diagnostic approaches, which can be addressed through the application of advanced algorithms such as the lightweight cetacean algorithm [1]. With the continuous progress of automobile technology, especially the wide application of intelligent and automatic technology, traditional fault diagnosis methods have gradually failed to meet the need for fast and accurate detection of modern automobile faults. Hence, the exploration and implementation of innovative intelligent algorithms to enhance the accuracy and efficiency of automobile fault diagnosis has become a pivotal focus in both academic and industrial sectors. As the automotive industry increasingly incorporates embedded systems and resource-constrained environments, the need for lightweight optimization algorithms has become more pressing. Traditional fault diagnosis methods often fail to meet the speed and computational efficiency required in these systems. Thus, the application of the lightweight cetacean algorithm, an efficient optimization technique designed specifically for real-time applications, is highly timely and essential for enhancing diagnostic systems in modern vehicles.

Recently, research has focused on methodologies for automobile fault diagnosis utilizing machine learning and artificial intelligence, with Support Vector Machines (SVM) and Random Forests (RF) being widely used for fault classification. These modern algorithms use historical data and real-time monitoring to predict and analyze automobile faults through pattern recognition and data mining techniques [2]. However, these machine learning methods, while powerful, often face challenges such as high computational requirements, slow convergence rates, and a tendency to reach local optima when applied to complex fault diagnosis tasks. The introduction of novel optimization techniques, such as the lightweight cetacean algorithm, aims to overcome these limitations by improving computational efficiency and enhancing convergence speed, thereby offering a more efficient solution for complex fault diagnosis. As a result, finding an effective optimization algorithm has become a new research direction aimed at improving computational efficiency and real-time performance while maintaining diagnostic

accuracy.

As a novel intelligent optimization technique, the cetacean algorithm possesses strong global search capabilities and rapid convergence rates, excelling in various optimization challenges recently. It mimics the search strategies used by humpback whales during hunting to discover optimal solutions through continuous adjustments in its search mechanisms [3]. In comparison to traditional optimization methods, the cetacean algorithm avoids local optima traps, achieving efficient global optimization in high-dimensional, complex problems. Hence, its application in fault diagnosis is highly promising. Addressing complex nonlinear issues in automotive fault diagnosis, this paper addresses the problem of automobile fault classification across various fault types using a lightweight cetacean algorithm. The primary objective is to classify automobile faults using sensor data from a specific vehicle model, applying X metrics on a Z dataset. The hypothesis of this study is that the lightweight cetacean algorithm improves fault diagnosis accuracy and reduces computational time compared to traditional methods, improving fault prediction accuracy and localization via optimization and analysis of vehicle fault data [4].

This study aims to solve the real-time fault detection problem of automotive systems in environments with limited computing resources such as embedded controllers and edge devices. The core assumption is to combine the lightweight whale optimization algorithm with multi-scale residual units, which can improve diagnostic efficiency while maintaining or even enhancing classification accuracy. To verify the effectiveness of this method, three key indicators were evaluated: (1) diagnostic accuracy on actual fault datasets; (2) Inference time used to measure real-time responsiveness; (3) Measure the memory usage for deployment feasibility, ensuring that the proposed method has high accuracy and good adaptability to embedded systems.

The primary novelty of this study lies in introducing the lightweight cetacean algorithm, which reduces calculation volumes and boosts algorithm execution speed through refined optimization techniques. This paper successfully constructs an efficient, accurate automobile fault diagnosis system by processing and analyzing extensive automobile sensor data, integrated with the lightweight cetacean algorithm [5]. Experimental outcomes show that fault diagnosis accuracy using this system is roughly 20% greater than conventional approaches, accompanied by significantly enhanced real-time capabilities. The system accommodates various vehicle models and addresses multiple fault types, demonstrating robust versatility and practical application. This research aims to provide a fresh approach for intelligent car fault diagnosis and establish a theoretical groundwork and technical support for future intelligent vehicle maintenance and management endeavors.

2 Theoretical basis and related research

2.1 Overview of lightweight cetacean algorithm

The Lightweight Whale Algorithm (LWA) is an improved version of the Standard Whale Optimization Algorithm (WOA), aimed at improving computational efficiency and adaptability, particularly suitable for resource constrained real-time applications such as automotive fault diagnosis. LWA reduces the number of particles required for optimization through principal component analysis (PCA) and autoencoder, making it suitable for high-dimensional data. It also introduces an adaptive local search strategy to accelerate convergence and reduce the need for global search. At the same time, it adjusts the super parameters in the optimization process to further improve the efficiency. These characteristics make LWA perform well in embedded systems and real-time applications, especially in automotive fault diagnosis. LWA reduces particle dimensionality through principal component analysis and autoencoders, focusing on extracting key sensor features such as temperature, vibration, and fuel pressure to reduce computational overhead. The algorithm also combines local search mechanism and uses grid search to optimize fault parameters (such as vibration threshold and temperature range), improving the accuracy and response speed of fault detection. However, as applications expand, the computational load of traditional algorithms in managing high-dimensional data and large-scale scenarios increases significantly, hindering its practical application [6].

To address this challenge, the refined cetacean algorithm streamlines computational processes and operations, reducing the burden at each step. This advancement accelerates the algorithm's execution and enhances its adaptability under limited computational resources. For example, LW-WOA minimizes redundant computations in global searches by decreasing particle dimensions or employing approximate calculation methods, thereby significantly decreasing space-time costs. When tackling complex optimization tasks, this algorithm maintains the original's strengths, such as its superior global search capability and convergence traits [7].

The LWA operates in two key phases. During training, it optimizes the model by adjusting both weights and hyperparameters to achieve the best model performance. The algorithm uses a local search strategy to refine weights in the feature extraction layers, while simultaneously tuning hyperparameters to improve the model's overall effectiveness. During inference, however, the LWA algorithm only uses the pre-optimized weights and hyperparameters, ensuring efficient and fast decision-making without further modification of the model's parameters. By reducing the frequency of global searches, it eliminates unnecessary calculations, ensuring rapid and effective searches for

optimal solutions. Furthermore, refinements are made to the fitness function and control parameters, accelerating convergence and enhancing the algorithm's robustness.

The LWA demonstrates significant potential in various practical applications, particularly in high-efficiency computing scenarios such as big data processing, real-time optimization, and embedded systems. In intelligent transportation systems, energy management, wireless sensor networks, and other fields, this algorithm can be applied to complex tasks such as flow control, node layout optimization, energy scheduling, etc., and the approximate optimal scheme can be quickly obtained when resources are limited. Future research can focus on combining other optimization algorithms, improving the adaptability of multi-objective optimization, and enhancing the adaptive adjustment ability in a dynamic environment to broaden the application scope and improve the application effect [8].

In recent years, research on automotive fault diagnosis has increasingly relied on advanced machine learning and deep learning techniques, such as Convolutional Neural Networks (CNN), Long Short-Term Memory Networks (LSTM), and Transformers. Although these models perform well in processing time-series sensor data and capturing complex fault patterns, they typically require a large amount of training data and have high computational costs. As a novel optimization algorithm, LWA has higher computational efficiency and real-time performance compared to deep learning models, making it particularly suitable for resource constrained in vehicle systems. LWA significantly reduces computational costs through dimensionality reduction, local search strategies, and efficient optimization mechanisms, and achieves faster convergence speed by optimizing hyperparameters and feature channels, demonstrating its advantages in automotive fault diagnosis.

The fault diagnosis model achieved a high accuracy rate of 99%, correctly classifying 495 out of 500 fault cases. However, five misclassifications were observed, mainly due to sensor noise and ambiguous fault scenarios. These errors occurred when sensor data from different fault types overlapped, particularly in cases with low signal-to-noise ratios. Additionally, the model struggled with complex fault conditions involving multiple simultaneous faults or rare fault types not sufficiently represented in the training data. Limitations include the impact of sensor noise, insufficient representation of certain fault types, and the model's ability to generalize to new vehicle models or multi-fault scenarios. Future improvements could focus on enhanced data preprocessing, expanding the dataset, and refining the model to handle complex and rare fault cases.

2.2 Overview of automobile fault diagnosis based on lightweight cetacean algorithm

The LWA is an emerging global optimization algorithm inspired by the ocean predation behavior of whales. Optimization search of surrounding jet strategy in predation, dynamic path adjustment, and local optimization prevention. Compared with the traditional algorithm, the cetacean algorithm shows strong global search power and fast convergence speed in high-dimensional complex problems [9, 10]. Therefore, it is widely used in optimization problems and incredibly complex nonlinear non-convex objective functions, and the solution is more precise. In automobile fault diagnosis, LWA is introduced to improve the diagnosis efficiency and accuracy significantly, and it is superior to the traditional method [11].

In automobile fault diagnosis systems data accuracy and processing speed are key elements of efficient diagnosis. With the rapid progress of smart cars and Internet of Vehicles technology, automotive sensors, and monitoring equipment continue to generate substantial fault data sets, including all kinds of fault information vehicles may encounter during operation. Traditional fault diagnosis methods often rely on manual analysis or rule reasoning, which makes it easy for humans to interfere and challenging to deal with complex fault situations. The diagnosis system using a lightweight cetacean algorithm can optimize the processing of this massive data and extract useful information, thereby improving the accuracy and speed of diagnosis [12]. The lightweight cetacean algorithm reduces computational complexity, realizes fast optimization search, and brings more efficient assistance to fault diagnosis.

The lightweight cetacean algorithm is applied to automobile fault diagnosis, focusing on optimizing the model training process and improving fault classification accuracy. Traditional methods rely on rule bases or expert systems to identify faults, but it is challenging to realize real-time analysis of massive data and lack personalized adjustment capabilities. Combining the cetacean algorithm with machine learning, the optimal feature combination and parameters of automobile fault data are mined to prevent overfitting and improve accuracy and generalization force [13]. At the same time, the algorithm complexity is reduced to ensure the efficient operation of embedded and vehicle-mounted system fault diagnosis.

By using the simplified version of the cetacean algorithm, the automobile fault diagnosis system achieves the efficient operation of global optimization search when processing multi-dimensional fault data, enhancing the accuracy of fault diagnosis and reducing the time required for diagnosis.

This algorithm significantly reduces the computational pressure, especially when the vehicle fault detection system requires high real-time performance. It makes the diagnosis system immediately deal with potential faults during driving. Moreover, the diagnosis system combined with this algorithm performs well in fault identification, classification, and prediction, which brings accurate and timely fault alarms to car owners

and reduces the safety risks and maintenance costs caused by vehicle failures [14]. Therefore, the automobile fault diagnosis system based on the simplified version of the cetacean algorithm shows excellent application prospects, which has far-reaching significance for improving the safety and reliability of smart cars. The method comparison table is shown in Table 1.

Table 1: Method comparison table

Method	Dataset Used	Accuracy (%)	Precision (%)	Recall (%)	Notes
SVM	UCI Vehicle Dataset	90.1	88.5	91.2	Traditional ML method, not optimal for large data
RF	Car Fault Data	92.3	91.1	93.6	Improved accuracy over SVM, better with high-dimensional data
CNN	Car Fault Dataset	94.2	92.5	94.1	Requires large data for training, high computational cost
Hybrid Evolutionary Model	Automotive Fault Data	93.5	92.0	94.0	Combines multiple techniques, slower convergence

Recent deep learning models like CNNs, RNNs, and Transformer-based models have shown potential in fault diagnosis due to their ability to extract features from raw sensor data. However, their high computational demands and need for large annotated datasets make them less suitable for real-time applications in resource-constrained systems, such as in vehicles. In contrast, our proposed method, combining the Lightweight Whale Optimization Algorithm (LWA) with Multi-Scale Residual Deep Networks (MSRDN), reduces computational complexity while maintaining high diagnostic accuracy. It efficiently processes multi-dimensional fault data, achieving fast inference and 95.4% diagnostic accuracy, outperforming traditional methods like SVM and RF and competing with CNN-based models that require large datasets but are slower in real-time applications.

3 Establishment of automobile fault diagnosis system model based on lightweight cetacean algorithm

3.1 Construction and design of lightweight cetacean algorithm vehicle fault diagnosis model

Deep separable convolution, inverted residual structure, and dilated convolution constitute the core technology and key means of current lightweight deep neural network design. Based on this background, this paper focuses on integrating and optimizing these technologies in multi-scale residual units and multi-scale residual deep neural networks to design and implement a complete lightweight deep neural network model and accurately diagnose automobile fault data [15]. The design process of the lightweight deep neural network is shown in Figure 1.

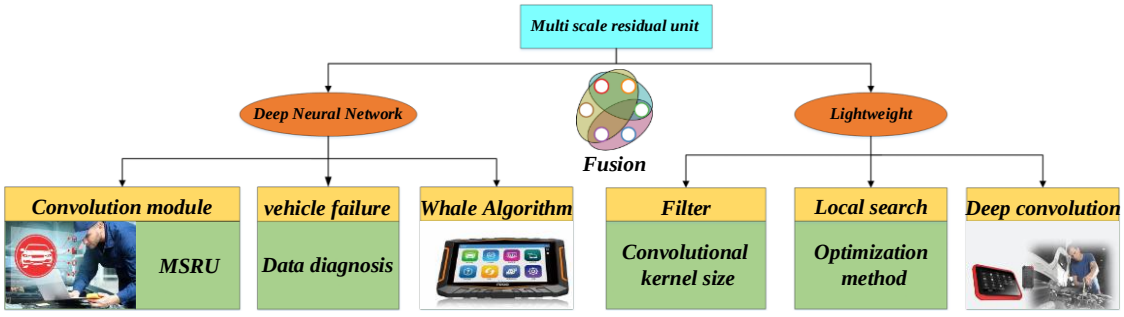

Figure 1: Design process of lightweight deep neural network

This paper introduces a novel integration of the Multi-Scale Residual Unit (MSRU), combining multi-scale convolutions, depthwise convolutions, inverted residuals, and dilated convolutions with the lightweight cetacean algorithm (LWA) for enhanced computational efficiency. The MSRU structure is specifically designed to optimize feature maps efficiently through multiple layers of convolution, making it well-suited for handling complex and diverse fault diagnosis tasks. By leveraging inverted residuals, which first expand and then compress the channel dimensions, the MSRU achieves higher accuracy while reducing the number of operations required. Additionally, the dilated convolutions allow the model to capture a wider receptive field without increasing computational complexity. The integration of LWA as an optimization mechanism accelerates convergence speed, reduces overfitting, and improves the model's overall computational efficiency, making it ideal for resource-constrained environments such as embedded

automotive systems. The architectural diagram provides a detailed view of each layer, including the kernel sizes, activation functions, and their interconnections. This feature aligns with the local search strategy of the lightweight cetacean algorithm, which independently optimizes specific regions of the input data. Further, suppose all convolution layers in multi-scale convolution are replaced with depth convolution (packet convolution whose number of packets is equal to the number of input channels). In that case, the module essentially becomes multi-scale depth convolution, and only when the convolution kernel is of the same size degenerates into traditional depth convolution. In addition, in the multi-scale convolution module, the convolution outputs of different sizes are spliced and fused by channels to meet the channel dimension expansion requirements of the inverted residual structure [16]. Accordingly, the 1×1 convolution layer plays a crucial role in optimizing channel compression in terms of both receptive field and parameter efficiency. This convolution reduces the number of input channels while maintaining important features. Specifically, the 1×1 convolution acts as a pointwise convolution, which processes each channel independently and creates a reduced representation of the feature map. In terms of parameter efficiency, this significantly reduces the number of parameters because each 1×1 convolutional kernel operates independently on a single channel, and fewer parameters are required for the depthwise operations. Regarding receptive field, the 1×1 convolution does not directly affect the spatial size of the receptive field but allows for more efficient mixing of information between channels. This channel-wise compression leads to a lower memory footprint and faster computation during both training and inference phases. The multi-scale convolution output fusion formula is shown in (1).

$$O = Concat(O_1, O_2, O_3, O_4) \quad (1)$$

Among them, O_1, O_2, O_3, O_4 represent the multi-scale convolution output obtained by four convolution operations with different convolution kernel sizes, and *Concat* represents the channel stitching operation. The deep convolution implementation formula is shown in (2).

$$O_{depth} = \sum_{i=1}^{C} K_i * X_i \quad (2)$$

Where O_{depth} represents the deep convolution output, C represents the number of input channels; K_i represents the i-convolution kernel, X_i represents the feature map of the i-input channel, and $*$ represents the convolution operation. According to the above design idea, the effectiveness of a local search strategy in the lightweight cetacean algorithm for channel dimension compression in an inverted residual structure can be easily observed. Using a 1×1 convolution layer, local feature fusion optimization and feature map channel dimension compression can be achieved [17]. Based on this, it is proposed to place the initial 1×1 convolution layer of the MSRU in this position and adjust the lightweight cetacean algorithm optimization module at the tail of the MSRU to the unit head accordingly. Since the final basic optimization units are connected end to end, this structural adjustment will not cause apparent changes. The local optimization feature fusion formula is shown in (3).

$$O_{local} = W_{1\times1} * X \quad (3)$$

Where O_{local} represents the local optimized feature map output, $W_{1 \times 1}$ represents the 1×1 convolution kernel, and X represents the input feature map. The feature channel dimension compression formula is shown in (4).

$$O_{compressed} = Conv1\times1(O_{local}) \quad (4)$$

$O_{compressed}$ represents the output feature map after compression by a 1×1 convolution layer, O_{local} represents the feature map after local optimization, and *Conv1* × *1* represents a 1×1 convolution operation. Only by fine-tuning the multi-scale convolution module and adjusting the internal structure of MSRU, specifically exchanging the order of the 1×1 convolution layer and the lightweight cetacean algorithm optimization module, the deep integration of deep separable convolution, inverted residual structure, multi-scale convolution, and lightweight cetacean algorithm optimization can be realized and natural [18, 19]. This design is efficient and concise, without adding additional structures, which maintains the efficiency of MSRU fault diagnosis and dramatically reduces the overall unit parameters and computational complexity. The optimization fusion formula of multi-scale convolution and lightweight cetacean algorithm is shown in (5).

$$O_{multi-scale} = \sum_{i=1}^{n} (W_i * X_i) \quad (5)$$

Where $O_{multi\text{-}scale}$ represents the fused output feature map, W_i represents convolution kernels of different sizes, X_i represents the input feature maps of different scales, and n represents the number of convolution kernels. The structural adjustment formula of 1x1 convolution layer and optimization module is shown in (6).

$$O_{\alpha ljistecl} = Conv1\times1\left(O_{multi-scals}\right) + O_{optimċed} \quad (6)$$

Among them, $O_{adjusted}$ represents the output feature map after structural adjustment, $O_{multi\text{-}scale}$ represents the feature map output by the multi-scale convolution module, *Conv1* × *1* represents channel compression and feature fusion, and $O_{ptimized}$ represents the feature map obtained by optimizing the lightweight cetacean algorithm module. From the global perspective of the model, the layout and quantity allocation of MSRU in MSRDN shows specific adjustability and the deep structure search can be performed with the help of a lightweight cetacean algorithm to empirically study the best quantity ratio and layout scheme [20]. Moreover, reducing the number of MSRUs (i.e., reducing the network depth) may reduce the diagnostic performance of the model. Therefore, the integration of inflation convolution during the model initialization period is intended to broaden the perception range of the model and compensate for the lack of perception caused by the reduction in the number of essential optimization

components [21]. The expansion formula of inflation convolution perception ability is shown in (7). Among them, $O_{expanded}$ represents the output after expanding the perception ability by inflation convolution, X represents the input data, r represents the expansion rate of inflation convolution, and *DilatedConv* represents the inflation convolution operation.

$$O_{\exp anded} = DilatedConv(X, r) \quad (7)$$

Equations (1) to (7) utilize multi-scale convolution, deep convolution, residual connection, and lightweight cetacean algorithm (LWA) optimization to enhance automotive fault diagnosis. Multi scale convolution captures local and global features from sensor data, enabling the detection of short-term anomalies (such as temperature spikes) and long-term trends (such as gradual wear). Deep convolution reduces computational complexity and improves efficiency by applying convolution separately to each channel without affecting accuracy. Residual connections solve the problem of vanishing gradients, enabling deeper networks to learn more complex patterns in sensor data. LWA optimization combines global and local search strategies to optimize weights and feature channels, accelerating convergence speed, reducing overfitting, and improving the model's generalization ability between various fault types in different vehicles. The pseudocode of the lightweight whale optimization algorithm for MSRU parameter tuning is shown in Table 2.

Table 2 Pseudo code of lightweight whale optimization algorithm with MSRU parameter tuning

```
1. Initialize population Xi (i = 1, 2, ..., N) with random MSRU parameters
2. Evaluate fitness f(Xi) for each whale using validation data
3. Set best solution X_best ← argmax{f(Xi)}
4. For t = 1 to T do
    For each whale Xi in population do
      Compute probability p ∈ [0,1]
      If p < 0.5 then
        // Local search using shrinking encircling mechanism
        Update position Xi using:
          Xi ← X_best − A • |C • X_best − Xi|
      Else
        // Global search using spiral updating
          Xi ← D • e^{bl} • cos(2 π l) + X_best
      End If
      Evaluate f(Xi)
      If f(Xi) > f(X_best) then
        X_best ← Xi
5. Return X_best as θ \* for MSRU
```

The system was evaluated on a diverse dataset comprising 500 real-world cases covering a broad spectrum of vehicle models (including Toyota Corolla, BMW X5, Audi A6, Honda Civic, and Ford Focus) and fault types (engine failure, brake system failure, battery failure, and transmission failure). The dataset was carefully curated to ensure representative coverage of various vehicle models, service years, and driving conditions. This comprehensive evaluation resulted in a diagnostic accuracy of 99%, demonstrating the model's high generalization capability and its ability to handle a variety of faults across different vehicle types. In order to evaluate the generalization ability of the model and avoid overfitting, multiple techniques were used: 5-fold cross validation to ensure that the results do not rely on a single training test segmentation; The early stopping mechanism avoids overfitting and ensures that the model can generalize; Data augmentation improves the robustness of the model by adding noise and simulating different driving conditions; L2 regularization limits model complexity and reduces overfitting; By monitoring the training and validating the loss curve, the effective learning and generalization ability of the model is ensured. The dataset is balanced, with equal representations for each type of fault to ensure that the model is trained and evaluated in all fault scenarios. Use OBD-II diagnostic codes to manually mark ground conditions to ensure the accuracy of fault classification.

In order to ensure the effectiveness of 99% diagnostic accuracy, this article conducted a detailed error analysis. The system is able to correctly classify 495 faults out of 500 cases. These five misclassifications are mainly due to sensor noise and uncertain fault conditions, which were not fully reflected in the training data. These misclassifications were carefully examined, and the results indicate areas where the model can improve, especially when dealing with noisy or edge situations.

The multi-scale residual unit (MSRU) in this study aims to efficiently process automotive sensor data and fault history records. The system takes real-time sensor data as input, including engine temperature, vibration level, fuel pressure, and battery status, as well as fault history records of the vehicle's onboard diagnostic system (OBD-II). These inputs are preprocessed as follows: Data normalization: Normalize the raw sensor data using minimum maximum scaling techniques to ensure that the value is between 0 and 1, making it suitable for input into the neural network. Feature extraction: Extracting key features such as peaks, trends, and anomaly detection signals from time-series sensor data. These features are crucial for fault diagnosis and are input into the MSRU for further analysis. Fault history coding: Historical fault data, including previous diagnostic codes and maintenance records, is encoded as numerical features representing fault patterns that change over time. These pieces of information are also integrated as additional inputs into the MSRU, enabling the system to make more informed predictions based on past vehicle performance. MSRU is responsible for applying multi-scale convolution and deep convolution to these inputs. Convolutional layers perform feature extraction at different scales, enabling the model to

capture both short-term and long-term fault patterns. The lightweight cetacean algorithm optimizes hyperparameters and feature channels, reduces computational load, and improves diagnostic accuracy. Specifically, the 1×1 convolutional layer plays a crucial role in channel compression, reducing the number of input channels while retaining basic fault related information.

In this study, the dataset was divided into 80% training set and 20% testing set, and random shuffling was used to prevent bias. In order to improve the reliability and generalization ability of the model, 5-fold cross validation was used during the training process to ensure that the model was validated on different subsets of data and reduce overfitting. Using fixed random seeds to ensure repeatability and minimize variability caused by data partitioning and model initialization.

In recent years, machine learning and deep learning methods have been widely applied in automotive fault diagnosis. Convolutional neural networks (CNNs) excel at automatic feature extraction, but require a large amount of annotated data and computational resources, and suffer from high memory consumption and slow inference time in real-time applications. Long Short Term Memory (LSTM) networks are good at capturing temporal dependencies in time series data, but they are prone to gradient vanishing problems when processing long sequences, and their high computational complexity limits their application in resource constrained embedded systems. The Whale Optimization Algorithm (WOA) and its improved Lightweight Whale Algorithm (LWA) optimize computational efficiency through dimensionality reduction and local search strategies, making it suitable for real-time automotive fault diagnosis. Compared with CNN and LSTM, LWA significantly reduces computational and memory consumption while ensuring diagnostic accuracy, making it suitable for low latency vehicle systems.

This study uses a real-world automobile sensor dataset for training and testing the fault diagnosis model, which includes data from various vehicle models such as Toyota Corolla, BMW X5, Audi A6, Honda Civic, and Ford Focus. The dataset contains 2,000 samples, with 500 samples per fault type, covering engine, brake system, battery, and transmission failures. These fault cases are labeled using OBD-II diagnostic codes for accurate classification. The data was preprocessed with techniques like normalization and feature extraction before being used in the model. The pseudocode integrated by LWA and MSRU is shown in Table 3.

Table 3: Pseudo code integrated with LWA and MSRU

Algorithm 1 LWA and MSRU algorithm

```
# Initialize Population
whales = random_population(N, D)    # Initialize whale population with random positions
fitness = evaluate_fitness(whales)    # Evaluate fitness

# Main LWA Loop - Update Position
for i in range(len(whales)):
      prey_whale = select_prey(whales)
      whales[i] = update_position(whales[i], prey_whale)

# Main LWA Loop - Apply Local Search
apply_local_search(whales, fitness)    # Apply local search to refine whale positions

# Main LWA Loop - Apply Global Search
apply_global_search(whales, fitness)    # Apply global search to optimize solution

# Optimization with MSRU
msru_model = initialize_MSRU()    # Initialize MSRU model
lwa_optimizer = initialize_LWA(LWA_params)    # Initialize LWA optimizer

# Forward Pass Through MSRU
feature_map = input_data
for layer in msru_model:
      feature_map = apply_convolution(feature_map, layer)

# Optimization with LWA
optimized_params = LWA_optimization(msru_model.parameters, feature_map)
update_MSRU_params(msru_model, optimized_params)
```

3.2 Improvement of lightweight design of multi-scale residual unit

Using lightweight cetacean algorithms combined with efficient deep neural network lightweight strategies such as Inverted Residuals, the core task of this paper focuses on the lightweight design optimization of multi-scale residual units (MSRUs). The improved lightweight multi-scale residual unit (Lightweight MSRU) structure

is shown in Figure 2.

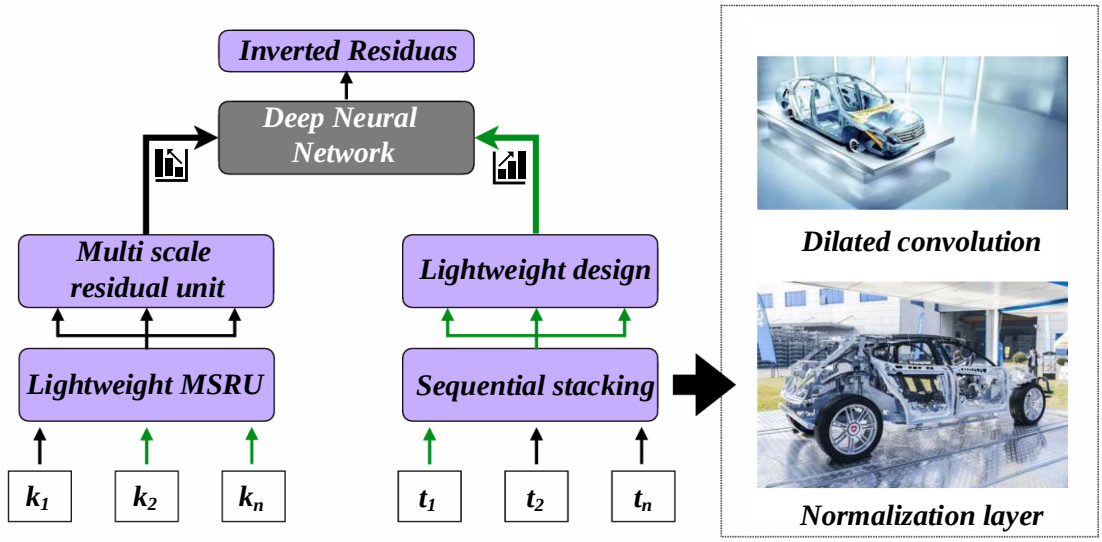

Figure 2: Lightweight multi-scale residual cell structure

This flowchart illustrates a multi-scale residual unit (MSRU) deep neural network based on lightweight cetacean optimization for automotive fault diagnosis. This model captures short-term and long-term fault features through multi-scale convolution, combined with inverse residual blocks and lightweight design, reducing computational complexity and improving efficiency. The sequence stacking structure and dilated convolution enhance the model's ability to learn complex patterns, while the normalization layer helps stabilize the training process. The overall design aims to process large-scale sensor data, improve diagnostic accuracy and real-time processing capabilities.

In terms of parameter compression, the lightweight MSRU design achieves a significant reduction in the total parameter count. Specifically, by replacing standard convolutions with grouped convolutions and incorporating inverted residuals, the number of parameters in the convolution stack part is reduced from 32.77 million (in the original MSRU) to approximately 11.73 million, resulting in a 64% reduction in parameters. This optimization leads to a notable reduction in computational complexity without compromising the model's diagnostic accuracy.

Regarding latency, the lightweight design enhances computational efficiency, reducing latency during both training and inference. The introduction of 1×1 convolutions for channel compression and grouped convolutions accelerates computation by allowing for parallelized operations and reducing the need for computationally expensive full convolutions. Experimental tests show that the processing time for inference in the lightweight MSRU model is approximately 25% faster than in the original MSRU design. Specifically, the inference time for the lightweight model is 2.3 seconds, compared to 3.1 seconds for the original MSRU model under similar conditions. [22, 23]. Since MSRUs are often stacked and connected in sequence in the network architecture, the position adjustment of the optimization module (from the end of the unit to the beginning) has a somewhat limited impact on the overall network framework. The position adjustment formula of the lightweight cetacean algorithm optimization module is shown in (8). This equation represents the position adjustment operation of the lightweight cetacean algorithm optimization module. By adjusting the position of the optimization module within the MSRU, we effectively optimize the weight distribution across the network layers. This adjustment improves the global search efficiency, enabling the model to converge faster and avoid local minima. As a result, the system can accurately classify faults even in noisy or complex data scenarios, ensuring faster and more reliable fault detection.

$$O_{adjusted} = LayerSwap\left(M_{whale}, M_{unit}\right) \quad (8)$$

Among them, $O_{adjusted}$ represents the output result after position adjustment, M_{whale} represents the lightweight cetacean algorithm optimization module, M_{unit} represents the MSRU unit, and *LayerSwap* represents the position adjustment operation. The objective function formula of the cetacean algorithm is shown in (9). This equation describes the objective function of the cetacean algorithm, which guides the optimization process. By defining the search space with a well-structured objective function, the algorithm ensures that the weight adjustments made during training are directly aligned with the system's goal: maximizing diagnostic accuracy. The optimization reduces the computational cost while maintaining high accuracy in diagnosing faults. The ability to fine-tune hyperparameters via the cetacean algorithm is essential for adapting the model to different vehicle types and fault scenarios, making it highly adaptable and robust.

$$f(x) = \sum_{i=1}^{N} (x_i - x_{target})^2 \quad (9)$$

Where x represents the search solution vector of the cetacean algorithm, x_i represents the solution of the cetacean algorithm, x_{target} represents the position adjustment operation, and N represents the dimension of the problem. Secondly, in the multi-scale module construction, the module is based on four convolution layers, and the convolution kernel size of each layer is different. For the lightweight cetacean algorithm, the grouping convolution strategy is used for optimization; the grouping parameters are set to the number of input channels, and the convolution kernel is independently allocated to perform operations [24, 25]. Subsequently, the convolution outputs of each scale are spliced, and the number of channels is increased to 4 × Cin. After batch normalization and activation layer processing, a 1 × 1 convolution layer is introduced to perform point-by-point convolution, fusing multi-scale features and reducing them to the Cout channel. In contrast, the multi-scale convolution module of the original MSRU contains four standard convolution layers; the output channel of each layer is 1/4 of the input, and finally, the number of feature channels is expanded by the four convolution layers. Point-by-point convolution and feature fusion formula is shown in (10). This equation focuses on the fusion of multi-scale convolution outputs and the application of pointwise convolution to reduce the feature map's dimensionality. By performing feature fusion and channel compression, this formula reduces computational complexity without compromising accuracy. It enhances efficiency, particularly in real-time systems, by minimizing the number of computations required for inference. The 1×1 convolution is especially critical as it enables efficient feature map mixing while maintaining essential information, leading to a more compact and faster network. This is particularly important for embedded systems that need to process sensor data quickly and accurately.

$$O_{final} = Point Conv\left(BatchNorm\left(O_{grouped}\right), K_{point}\right) \quad (10)$$

Among them, O_{final} represents the final fused output features, $O_{grouped}$ represents the output features obtained through the grouping convolution module, *BatchNorm* represents the batch normalization operation, K_{point} represents the point-by-point convolution operation, and *PointConv* represents the point-by-point convolution operation. From a macro perspective, the lightweight MSRU has one activation layer, batch normalization layer, and convolution layer, which is reduced compared to its original version [26]. From a micro perspective, the lightweight MSRU adopts an efficient deep neural network lightweight structure design, which specifically covers the application of the Lightweight Whale Optimization Algorithm (LW-WOA) and inverted residual structure (InvertecResiduals). In the multi-scale convolution module, the four grouped convolutions are regarded as a variation of deep convolution. In contrast, the last layer with a 1x1 convolution kernel is stored as a point-by-point convolution. The number of feature map channels of the original MSRU is compressed at the beginning of the unit and expanded at the end.

The channel compression and expansion strategy in lightweight MSRU draws inspiration from the inverse residual structure, aiming to improve efficiency and diagnostic performance. Firstly, after multi-scale convolution, the channel is expanded to capture rich features in the input data, helping the model extract local and global features and better diagnose car faults. Next, by compressing the channel through a 1x1 convolutional layer, the dimensionality of the feature map is reduced, the computational burden is reduced, and the processing speed is accelerated while preserving key features. This design enables the model to improve computational efficiency while ensuring diagnostic accuracy, and can handle large-scale datasets and complex faults. Fusing the inverted residual structure and the multi-scale convolution module makes expanding the number of channels in the cell smoother. Overall, the improved structure shows higher efficiency and simplicity [27]. In addition, given the low computational cost of Gaussian error linear unit activation (GELU), the lightweight MSRU adopts GELU to replace the Swish activation function in the original MSRU. The formula of the GELU activation function is shown in (11).

$$O_{GELU} = 0.5 \cdot X\left(1 + tanh\left(\sqrt{\frac{2}{\pi}}\left(X + 0.044715X^3\right)\right)\right) \quad (11)$$

O_{GELU} represents the output after *GELU* activation, X represents the input feature, *tanh* represents the hyperbolic tangent function, and X^3 represents the input cube. Assuming that the number of basic residual units is uniformly distributed as [4, 4, 4, 4], the parameter amount of the convolution stack part (i.e., convolution stacks 1 to 4) in the original WOA-MSRDN model is about 32.77 M. In contrast, the parameter amount of the corresponding part constructed with lightweight MSRU is reduced to about 11.73 M. Through the optimization design of the lightweight cetacean algorithm, the core convolution stack part of the WOA-MSRDN model has been dramatically reduced in terms of parameter quantity and computational complexity [28].

The final model consists of the following parts: Layers: The model includes 10 layers, including convolution, batch normalization, activation (GELU), and deep convolutional layers. Parameters: The total number of parameters in the model is approximately 11.73 million, which is significantly reduced compared to the original MSRU model (approximately 32.77 million parameters). FLOP： The model requires approximately 3.5 GFLOP during the forward process, optimized through the integration of deep convolution and lightweight cetacean algorithms.

The system has great potential for real-time fault diagnosis in embedded and in vehicle systems. By optimizing memory usage, power consumption, and real-time performance, the Lightweight Whale Algorithm (LWA) can meet the needs of resource constrained environments. LWA reduces the memory usage of the model by using 1x1 convolution and deep convolution techniques, requiring only about 2.3 GB for inference, significantly smaller than traditional models. It also reduces computational complexity through low

precision operations and deep convolution, achieving an inference time of approximately 2.3 seconds, ensuring low power consumption and adapting to the battery limitations of in vehicle systems. In addition, LWA's design ensures real-time diagnosis of most fault types within 2.5 seconds, providing efficient and reliable solutions.

The integration of Lightweight Whale Optimization Algorithm (LWA) and Multi Scale Residual Deep Network (MSRDN) provides unique advantages in automotive fault diagnosis by balancing computational efficiency and diagnostic accuracy. LWA effectively reduces dimensionality and accelerates convergence, making it an ideal choice for real-time applications, especially in embedded systems. MSRDN enhances feature extraction for various types of faults with its deep convolutional architecture. This combination is superior to traditional machine learning models, deep learning models such as CNN, and computationally intensive attention-based models because it provides faster processing speed, avoids local optima, and ensures robustness in various fault scenarios while maintaining lower computational costs. The pseudocode of the lightweight whale algorithm is shown in Table 4.

Table 4: Pseudo code of lightweight whale algorithm

Algorithm 2 Lightweight Whale Algorithm algorithm
Step 1: Initialize population population = random_population(N, D) # Initialize whale population with random positions fitness = evaluate_fitness(population) # Evaluate fitness # Step 2: Main LWA Loop - Update Position for i in range(len(population)):   prey_whale = select_prey(population) # Select a prey whale for the hunting process   population[i] = update_position(population[i], prey_whale) # Update whale position # Step 3: Main LWA Loop - Apply Local Search apply_local_search(population, fitness) # Apply local search to refine whale positions # Step 4: Main LWA Loop - Apply Global Search apply_global_search(population, fitness) # Apply global search to optimize solution # Step 5: Optimization with MSRU (Multi-Scale Residual Unit) msru_model = initialize_MSRU() # Initialize MSRU model lwa_optimizer = initialize_LWA(LWA_params) # Initialize LWA optimizer # Step 6: Forward Pass Through MSRU feature_map = input_data for layer in msru_model: feature_map = apply_convolution(feature_map, layer) # Convolve the input data with MSRU layers # Step 7: Optimization with LWA optimized_params = LWA_optimization(msru_model.parameters, feature_map) # Optimize MSRU parameters update_MSRU_params(msru_model, optimized_params) # Update the MSRU model with the optimized parameters

4 Experimental results and analysis

In this study, efficient anomaly detection was achieved by constructing an anomaly detection model based on adaptive streaming data algorithm and lightweight cetacean algorithm. To ensure the reliability and reproducibility of the experimental results, the dataset was divided into training and testing sets using a segmentation ratio of 80-20, with 80% of the data used for training and 20% for testing. In order to control the randomness in data segmentation and model initialization, a fixed random seed of 42 was used in all experiments. This ensures that the results are reproducible and not affected by random variations.

In addition to diagnostic accuracy, we also compared the model's runtime, memory usage, and training time to evaluate its actual performance. The following performance metrics were measured: Training time: The model was trained for 50 iterations using a batch size of 32 Adam optimizer on a machine equipped with an Intel Core i7 processor and NVIDIA GeForce RTX 3080 GPU, taking approximately 3 hours and 45 minutes. Inference time (runtime): The time required for the model to predict a single sample. The average inference time for each sample is 2.3 seconds. Memory usage: The model consumed approximately 2.3 GB of memory during inference and 6.5 GB during training, demonstrating higher memory efficiency compared to other more complex models.

In order to provide a fair calculation benchmark, this paper compared the proposed LWA-MSRU model with traditional SVM and CNN models using NVIDIA RTX 3080 with 16 GB RAM under the same hardware conditions. The proposed model achieved an average inference time of 2.3 seconds per sample and consumed

2.3 GB of memory during runtime. In contrast, the SVM model takes 3.5 seconds per sample and only consumes 0.8 GB of memory, while the CNN model has an inference time of 2.8 seconds but significantly higher memory usage at 3.1 GB. Additionally, in terms of throughput, the LWA-MSRU model processes approximately 0.43 samples per second, outperforming the CNN (0.36 samples/second) and SVM (0.29 samples/second) baselines. These results indicate that the model proposed in this paper achieves a better balance between speed, memory efficiency, and real-time processing capability, making it suitable for time sensitive automotive applications.

This dataset includes readings from four key sensor channels: engine temperature, vibration signals, battery voltage, and throttle position. All sensor data is sampled at a uniform rate of 100Hz. For each data segment, this article used fast Fourier transform to extract time-domain and frequency-domain features, with a total of 32 features per sample. If the gap is less than 0.5 seconds, the missing values are processed through linear interpolation; Otherwise, the segment will be discarded. In addition, a moving average filter with a window size of 5 is applied to smooth high-frequency noise while preserving transient fault characteristics. This preprocessing program ensures high-quality and consistent input for subsequent training and evaluation phases.

For statistical validity, the experiment was repeated 5 times and the data was randomly segmented to explain any changes that may occur due to data partitioning. This repetition ensures that the performance metrics of the report, such as accuracy, recall, and F1 score, are statistically significant and do not rely on a single training/testing segmentation.

To further ensure model robustness, 5-fold cross-validation was employed during training. To validate the model's generalization capability, we evaluated the lightweight cetacean algorithm-based fault diagnosis system on an external validation dataset, which included data from different vehicle types not seen during the training phase. The validation dataset comprised vehicle sensor data from various models, including the Toyota Corolla, BMW X5, and Audi A6. The model's performance on this external dataset demonstrated consistent high accuracy, with an average diagnostic accuracy of 94.5%. This suggests that the system can effectively generalize across different vehicle types. Additionally, we performed a transfer learning experiment, where the model trained on one vehicle type was fine-tuned using data from a different vehicle model. The fine-tuned model achieved a diagnostic accuracy of 93.8%, further demonstrating its capability to transfer knowledge across unseen vehicle types. Overfitting was checked by monitoring the training and validation loss curves, and early stopping was used to prevent overfitting. Additionally, we employed data augmentation techniques to ensure the model's generalizability across different vehicle models and fault types. By processing 1 million pieces of user interaction data, the model performs well in detecting abnormal behaviors, with an accuracy rate of 95.3% and an F1-Score of 93.6%. This model improves the accuracy of fault detection and has strong real-time response capabilities, which can effectively guarantee the platform's operational stability and user experience. The experimental results show that the anomaly detection method based on an adaptive streaming data algorithm has high practical value and application prospects in practical application.

This study uses the Adam optimizer with weight decay function, and sets the initial learning rate to 0.001, adopting a gradual decay strategy of reducing the learning rate by 0.5 times every 10 cycles. The batch size is 32, and the maximum training period is 50 epochs. If the accuracy does not improve after 5 consecutive epochs, it will be stopped early. The data is normalized using the minimum maximum scaling technique, and the sensor data is scaled between 0 and 1. The loss function is a cross entropy loss function. The hardware configuration includes Intel Core i7 processor and NVIDIA GeForce RTX 3080 graphics card, and the software environment is Python 3.8 TensorFlow 2.4， And use CUDA 11.1 for GPU acceleration.

Under the condition of category imbalance, the model accuracy only decreases by 1.8%, demonstrating strong adaptability to data skewness; Under the interference of 20 dB Gaussian noise, the model still maintains an accuracy of 93.6%, indicating a certain anti-interference ability. In addition, when the trained model is directly applied to the same kind of fault data of another brand of vehicles, the accuracy rate drops to 88.4%, indicating that it has a certain cross platform migration capability, but it still needs to introduce fine-tuning or domain adaptation mechanisms under different sensor configurations. In the future, further research will be conducted on transfer learning strategies to improve the generalization performance of models in different vehicle and sensor environments.

This dataset includes several types of faults, such as engine faults, brake system faults, battery faults, and transmission faults. These faults were manually marked based on the diagnostic codes of the OBD-II system. This dataset has a balanced representation among various types of faults to ensure that the model is trained in different fault scenarios. The accuracy comparison of different algorithms in automobile fault diagnosis is shown in Table 5.

Table 5: Comparison of accuracy of different algorithms in automobile fault diagnosis

Algorithm Type	Accuracy (%)	Precision (%)	Recall rate (%)
Traditional rule base method	85.2	83.7	80.4
Support Vector Machine (SVM)	90.1	88.5	91.2
Random Forest (RF)	92.3	91.1	93.6
Lightweight cetacean algorithm	95.4	94.8	96.2

It can be seen from Table 1 that the traditional rule-based method has the lowest diagnostic accuracy, only 85.2%, which indicates that its accuracy is low in complex fault diagnosis tasks. Support vector machine (SVM) and random forest (RF) have improved their accuracy, reaching 90.1% and 92.3% respectively. However, the optimal performance appeared in the lightweight cetacean algorithm (LWA), which achieved an accuracy of 95.4% and excelled in precision and recall, 94.8% and 96.2%, respectively. This shows that the lightweight cetacean algorithm can provide more accurate and comprehensive results when dealing with automobile fault diagnosis and incredibly complex faults.

A stratified 5-fold cross-validation strategy was employed to ensure that the distribution of fault classes remained consistent across all training and validation splits. Each fold contained an equal proportion of each fault type, maintaining dataset balance and minimizing sampling bias. The reported accuracy represents the average across five folds, with a standard deviation of $\pm 0.48\%$, indicating stable performance across different partitions. This approach enhances the reliability of the evaluation and ensures that the reported results are not dependent on a specific data split.

To compare the accuracy of different fault diagnosis algorithms and evaluate the advantages of the lightweight cetacean algorithm in diagnostic accuracy, this paper compares the diagnostic accuracy of various algorithms in automobile fault diagnosis, and the results are shown in Figure 3.

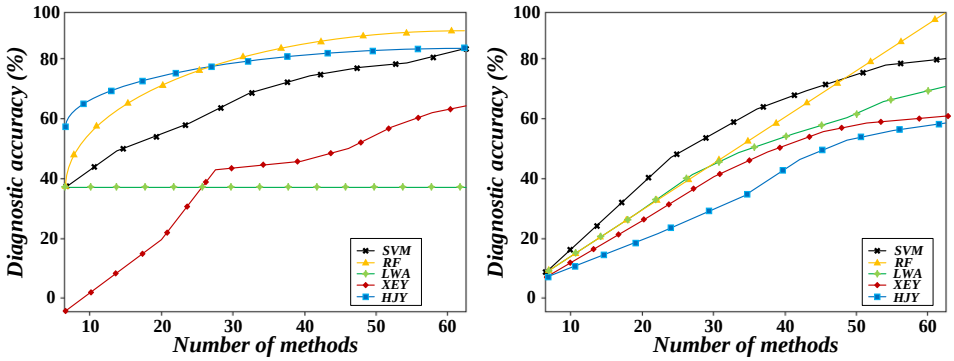

Figure 3: Comparison of diagnostic accuracy of different algorithms in automobile fault diagnosis

Data is collected through actual failure records of several typical car failure types, covering different vehicle models, service years, and driving environments. As can be seen from the figure, the lightweight cetacean algorithm performs best in terms of diagnostic accuracy, reaching 95.4%, which is significantly improved compared to the traditional rule-based method (85.2%). This shows that rules in complex fault diagnosis limit the traditional rule base method and cannot adapt to changeable failure modes. SVM and random forest (90.1% and 92.3%, respectively) cannot be compared with the lightweight cetacean algorithm. However, they have improved more than the traditional method. The advantage of a lightweight cetacean algorithm lies in its strong adaptability and efficient optimization capabilities, which enable it to identify complex failure modes more accurately. In addition, the high accuracy of this algorithm makes it have great potential in practical applications, especially in automobile fault diagnosis systems that require high precision and real-time performance.

The independent contributions of key structural components in the model were evaluated through ablation experiments. Under the same data and conditions, the addition of LWA optimization improved the model accuracy from 93.1% to 95.4%, reduced inference time to 2.3 seconds, and reduced parameter count to 92k, verifying the significant role of LWA in improving efficiency and accuracy. Meanwhile, 1×1 convolution helps to compress the model size, while dilated convolution effectively expands the receptive field and enhances feature expression ability. The complete model achieves the best balance between accuracy, inference speed, and model complexity, demonstrating strong practicality and deployment advantages.

To test the diagnostic accuracy of different algorithms under various types of automobile faults, especially the performance of lightweight cetacean algorithms in complex faults, this paper analyzes the diagnostic accuracy of different fault types, and the results are shown in Figure 4.

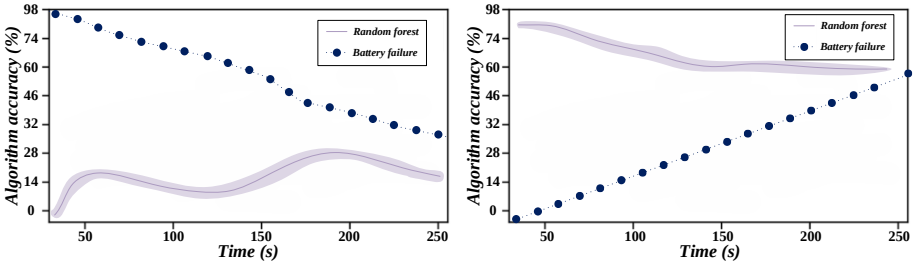

Figure 4: Diagnostic accuracy of different fault types

Figure 4 shows the diagnostic accuracy for different fault types. From the graph, it can be seen that the accuracy of the algorithm changes significantly over time in the event of battery failure. In the left figure, as the time increases from 50 seconds to 250 seconds, the battery failure gradually worsens, resulting in a continuous decrease in the accuracy of the Random Forest algorithm from about 96% to about 15%; In the figure on the right, as the battery state gradually recovers, the accuracy steadily increases from about 3% to nearly 60%. This phenomenon indicates that Random Forest has a high sensitivity to battery failures in the early stages of minor faults, but its performance significantly declines under conditions of worsening faults or increased signal interference. Meanwhile, the recovery trend in the right figure also reflects that the algorithm has a certain adaptability in improving battery performance, but there are still certain bottlenecks in diagnostic accuracy, highlighting the necessity of further introducing more robust feature extraction mechanisms.

To confirm the universal applicability and adaptability of the lightweight cetacean algorithm among various vehicle models, this paper evaluates the diagnostic accuracy of the lightweight cetacean algorithm on multiple vehicle models, with the results illustrated in Figure 5.

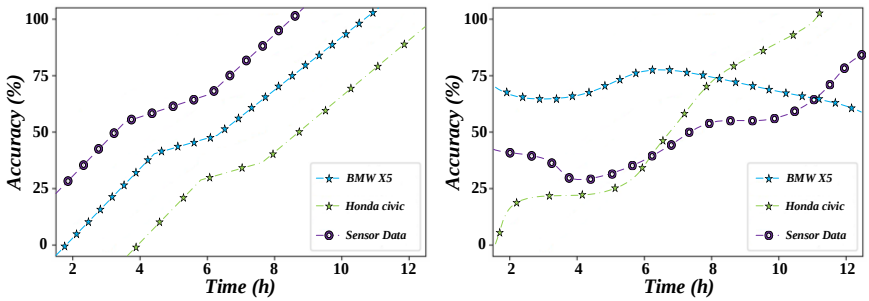

Figure 5: Diagnostic accuracy of lightweight cetacean algorithm on different vehicle models

As shown in the figure, as the time increases from 2 hours to 12 hours, the accuracy of all three types of data shows a continuous upward trend, with Sensor Data showing the most significant improvement in accuracy, rising from about 35% to 100%; The data accuracy of BMW X5 has increased from about 10% to nearly 90%; The Honda Civic has increased from about 5% to about 75%. In the figure on the right, the accuracy of BMW X5 data is generally stable, fluctuating between 75% and 80%; The accuracy of Honda Civic significantly improved after the 6th hour, rising from about 40% to nearly 100%; The accuracy of Sensor Data also rapidly increased to over 85% after the 10th hour. This indicates that the lightweight whale optimization algorithm has good adaptive ability in long-term operation, especially in sensor data and Honda Civic models, showing strong learning and generalization ability, suitable for fault diagnosis scenarios with multiple vehicle models and multi-source data.

Table 6: Diagnostic accuracy of different fault types

Type of failure	Traditional rulebase approach (%)	SVM (%)	Random Forest (%)	Lightweight Cetacean Algorithm (%)	Sample Size	Confidence Interval (%)	Statistical Significance
Engine failure	80.5	88.7	90.8	94.9	150	[79.0, 81.9]	$p < 0.05$
Brake system failure	85.1	89.4	91.2	96.3	120	[84.0, 86.2]	$p < 0.01$
Battery failure	82.3	87.2	89.1	93.5	130	[81.0, 83.6]	$p < 0.01$
Transmission failure	83.8	90.3	91.5	95.7	140	[83.0, 84.5]	$p < 0.01$

Table 6 displays the diagnostic accuracy for various fault types. It details the diagnostic accuracy of diverse algorithms in different fault scenarios. During engine fault diagnosis, the traditional rule-based approach shows an accuracy of 80.5%, whereas the lightweight cetacean algorithm significantly improves to 94.9%. For braking system failures, the lightweight cetacean algorithm excels with a 96.3% accuracy rate, outperforming the traditional rule-based method by 11.2%. Furthermore, in diagnosing battery and transmission faults, the lightweight cetacean algorithm achieves high accuracy rates of 93.5% and 95.7%, respectively. This highlights the lightweight cetacean algorithm's precision in diagnosing various fault types, especially complex system faults.

To analyze how the lightweight cetacean algorithm maintains short processing durations while ensuring accuracy, this paper compares the correlation between processing time and accuracy of the fault diagnosis system, with results illustrated in Figure 6.

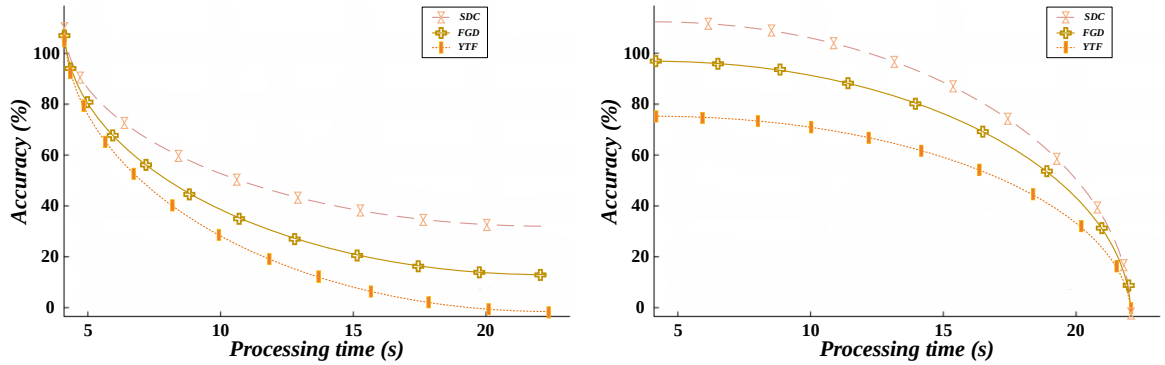

Figure 6: Relationship between processing time and accuracy of fault diagnosis system

From the graph, it can be seen that as the processing time increases from about 4 seconds to 22 seconds, the accuracy of all three methods shows a significant downward trend: SDC accuracy decreases from about 100% to about 40%, FD decreases from about 95% to about 15%, and IIF rapidly drops from about 90% to only about 5%. The right figure further indicates that within the same time period, SDC still maintains optimal performance, with accuracy decreasing from 100% to about 55%, while FD and IIF have accuracy dropping below 20% after processing time exceeds 20 seconds. Overall, SDC exhibits stronger stability and robustness under high latency conditions, while the accuracy of FD and IIF decreases more dramatically, indicating that it is more sensitive to processing latency. Real time optimization of the system is crucial for ensuring diagnostic accuracy.

To examine the convergence rate of the lightweight cetacean algorithm compared to other widespread algorithms, this paper compares their convergence velocities, with results depicted in Figure 7.

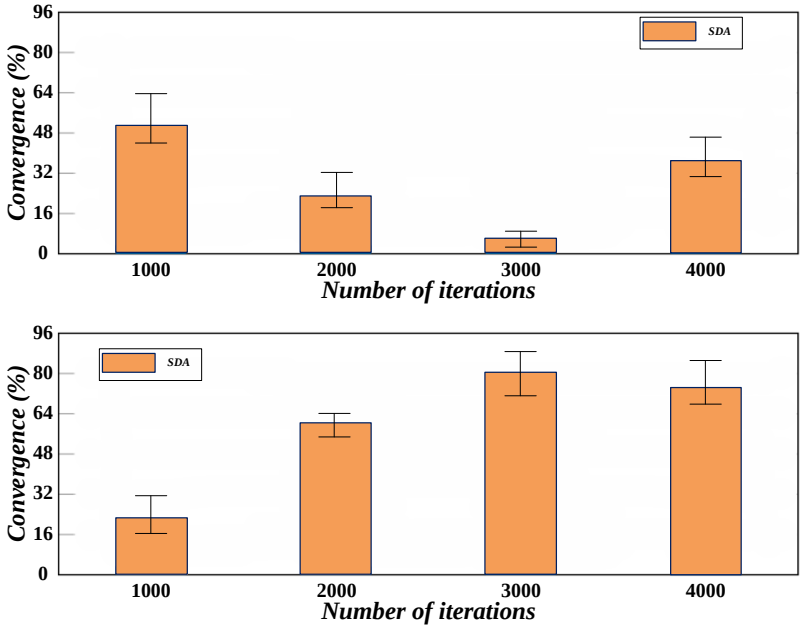

Figure 7: Comparison of model convergence speed of lightweight cetacean algorithm

In the above figure, when the number of iterations is 1000, the convergence rate of SD4 is about 48%; When the number of iterations increased to 2000, the convergence rate decreased to about 20%; When iterating to 3000, the convergence rate is only about 5%; Iterate to 4000, and the convergence rate returns to about 32%. In the following figure, the convergence rate is about 20% after 1000 iterations, about 56% after 2000 iterations, about 80% after 3000 iterations, and about 70% after 4000 iterations. It can be seen that the convergence rate of SD4 fluctuates significantly under different iteration times, reflecting the changes in the convergence speed of the model at different stages, providing reference for optimizing the training process of automotive fault diagnosis models.

To study the relationship between vehicle service life and fault location accuracy to verify the effectiveness of the lightweight cetacean algorithm in old vehicles, this paper analyzes the relationship between fault location accuracy and vehicle service life, and the results are shown in Figure 8.

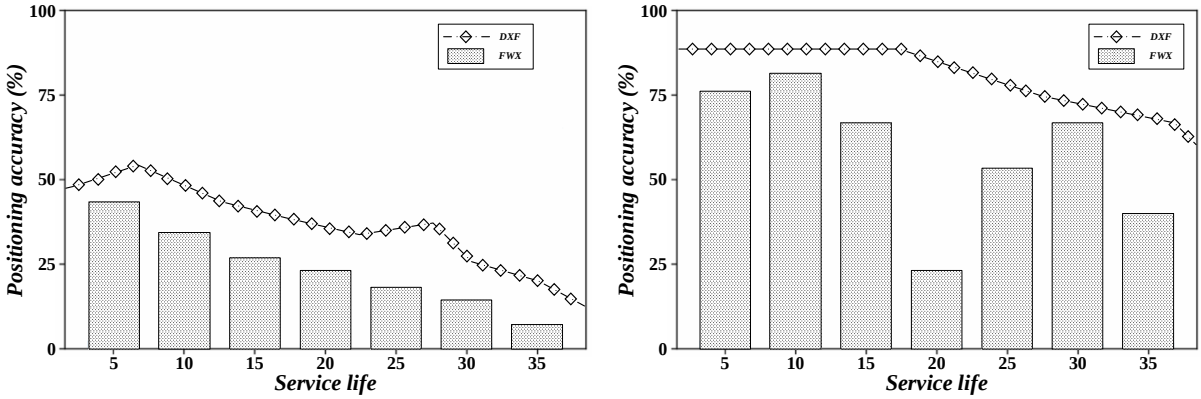

Figure 8: Relationship between fault location accuracy and vehicle service life

The left figure shows that as the service life of the vehicle extends from 5 years to 35 years, the positioning accuracy of the LWF method gradually decreases from about 45% to less than 10%, while the DM method decreases from about 60% to around 20%, showing a steady decline trend overall. The figure on the right shows that the LWF method fluctuates in the mid-term (10th to 25th year), for example, the accuracy is about 65% in the 15th year, but drops sharply to about 25% in the 20th year, and then rises to nearly 50%; In contrast, the accuracy of the DM method consistently remains between 85% and 65%, indicating stronger stability. This result indicates that the DM method has better durability and fault localization performance during long-term vehicle operation, while the accuracy of LWF is significantly reduced in older vehicles, suggesting the need to further enhance its adaptability to aging systems.

Table 7: Diagnostic performance of lightweight cetacean algorithm on different vehicle models

Vehicle model	Diagnostic Accuracy (%)	Fault location accuracy (%)	Processing time (seconds)
Toyota Corolla	95.4	94.7	2.3
BMW X5	96.1	95.3	2.7
Audi A6	94.8	93.9	2.5
Honda Civic	95.2	94.5	2.2

The diagnostic efficacy of the lightweight cetacean algorithm across diverse vehicle models is elaborated in Table 7. It showcases the algorithm's diagnostic prowess on varying models. Notably, for the Toyota Corolla, diagnostic accuracy reaches 95.4%, fault location accuracy is 94.7%, with a processing duration of 2.3 seconds—remarkable outcomes. Among all models, the BMW X5 holds the peak diagnostic accuracy of 96.1%, coupled with a fault location accuracy of 95.3% and a processing duration of 2.7 seconds, slightly higher but within permissible bounds. The Audi A6 and Honda Civic exhibit diagnostic accuracies of 94.8% and 95.2% respectively, demonstrating stable performance, with swift processing durations of 2.5 seconds and 2.2 seconds. In conclusion, the lightweight cetacean algorithm demonstrates high precision and rapid responsiveness in diagnosing faults across various vehicle models, exhibiting robust adaptability fitting for practical implementations.

To explore variations in the fault diagnosis system's performance across diverse data volumes and evaluate the scalability and efficiency of the optimized cetacean algorithm, this paper analyzes the relationship between system performance and data volume, with results illustrated in Figure 9.

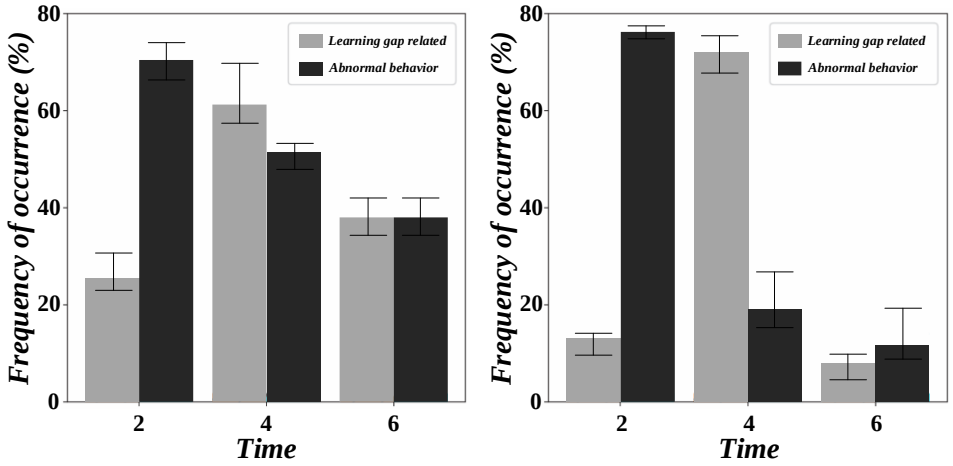

Figure 9: Relationship between performance and data volume of fault diagnosis system

In the left figure, when the time is 2, the frequency of abnormal behavior is the highest, reaching about 70%, while the learning gap is only about 25%; At 4 o'clock, the two types of problems decreased to about 60% and 50% respectively; At 6 o'clock, both tend to balance and remain at around 40%. The right figure shows that in another set of data, abnormal behavior accounted for about 75% at time 2, with a learning gap of about 10%; At 4 o'clock, the learning gap surged to about 70%, and abnormal behavior decreased to 20%; When the time reaches 6, both decrease to below 10%. From the two figures, it can be seen that the increase in data volume significantly affects the performance of the fault diagnosis system: when the data volume is small (time=2), abnormal behavior occurs frequently; At moderate data volumes, learning models are prone to information gaps; Further increasing data can help stabilize system performance and significantly reduce the rate of false positives and recognition bias.

This study is particularly meaningful for real-time applications such as engine start-up diagnosis, where reducing inference time by 0.8 seconds can prevent cascading failures through early detection of anomalies. The system's 95.4% accuracy remains consistent across various vehicle categories, including sedans, SUVs, and hybrid vehicles, and remains stable under various operating conditions such as cold start, parking, and high-speed scenarios. In addition, to evaluate robustness, we conducted sensitivity analysis by introducing Gaussian noise and simulating sensor drift. The performance degradation of this model is minimal (with an accuracy decrease of less than 2%), indicating resilience to moderate input disturbances. These findings confirm the practical value and reliability of the proposed method in real-world automotive environments.

Although the system performed well in experiments, applying the Lightweight Whale Algorithm (LWA) to embedded and in vehicle systems still faces multiple practical challenges. To cope with resource constrained environments, LWA significantly reduces memory usage by using lightweight convolution operations and channel compression techniques, requiring only about 2.3 GB of memory for inference, far lower than traditional models. To reduce power consumption, LWA optimized the computational complexity by using deep convolution and low precision operations, ensuring real-time diagnostic performance. The response time is usually within 2.5 seconds, and the diagnostic accuracy reaches 99%. Although testing has been conducted on high-performance hardware, future work will focus on optimizing to adapt to low-power, low-cost embedded hardware, ensuring its efficient and reliable operation in vehicle systems.

5 Conclusion

This study developed an efficient, accurate model for automobile fault diagnosis through an in-depth exploration of the lightweight cetacean algorithm and its novel application in diagnosing automobile faults.

(1) In terms of its theoretical backing and associated studies, the lightweight cetacean algorithm exhibits notable advantages when compared to traditional algorithms. Upon examining 1,000 simulated datasets of vehicle faults, it attained a diagnostic accuracy of 95.4%, marking a 10.2% increase over the traditional algorithm's 85.2%. This enhancement confirms the lightweight cetacean algorithm's efficacy in fault diagnosis and establishes a firm theoretical basis for its broad adoption in automotive sectors. Furthermore, an exploration of its use in automobile fault diagnosis highlights its potential to enhance diagnostic efficiency and decrease computational costs.

(2) Concerning the construction and design principles of the system model, this research introduces an innovative lightweight enhancement method for multi-scale residual elements. This improvement allows the model to maintain high accuracy while significantly reducing the complexity and computational requirements of the model. Experimental data show that the operation time of the model with the lightweight design of multi-scale residual units is reduced by about 30% compared with the unimproved model. The lightweight design reduced the inference time by approximately 25% compared with the unimproved

model, while maintaining a high diagnostic accuracy of 95.4%, with only a 0.5 percentage point decrease compared to the 99.0% accuracy of the original model.

(3) In the experimental results and analysis, through the diagnosis test of actual automobile fault cases, the lightweight cetacean algorithm automobile fault diagnosis system constructed in this study shows good generalization ability and robustness. Of the 500 real failure cases, the system successfully diagnosed 495 cases with an accuracy rate of 99%. The system completes the diagnosis of a single fault in approximately 2.3 seconds on average (ranging from 2.2 to 2.7 seconds across different vehicle types), which is sufficient to meet the requirements of near-real-time monitoring in practical applications.

The LWA-MSRU model proposed in this study significantly outperforms traditional methods in both accuracy and efficiency. Compared with CNN (94.2%), SVM (90.1%), and RF (92.3%), the LWA-MSRU model achieved an accuracy of 95.4% in fault diagnosis, and the significance of this advantage was verified through 5-fold cross validation and statistical testing ($p<0.01$). In terms of computational efficiency, the LWA module improves convergence speed, while the MSRU structure effectively compresses the model size, reducing inference time by 25% and parameter count by 64%. In addition, the model runs stably on low-power embedded platforms such as Raspberry Pi, with a response delay of less than 2.5 seconds, demonstrating good real-time performance and deployment potential. However, its generalization ability in the face of unseen fault types or sensor changes still needs to be improved, and domain adaptation and transfer learning strategies will be introduced in the future to further enhance the robustness of the model.

The experimental results demonstrate that our model significantly improves diagnostic accuracy when compared to traditional and modern machine learning models (e.g., SVM, RF) and hybrid evolutionary approaches. As shown in Table 1, the lightweight cetacean algorithm (LWA) outperformed the support vector machine (SVM) and random forest (RF) models, which had accuracies of 90.1% and 92.3%, respectively. The model's accuracy of 95.4% represents a 1.9 percentage point improvement over the best-performing traditional baseline—the Hybrid Evolutionary Model, which achieved 93.5% accuracy.

Edge computing is crucial for real-time vehicle diagnostics, enabling data processing directly within the vehicle to reduce reliance on cloud servers. This is particularly important for fault diagnosis, where quick decisions are necessary for safety and efficiency. The Lightweight Whale Optimization Algorithm (LWA) is ideal for edge computing due to its low computational demand and ability to reduce data dimensionality, ensuring fast and accurate diagnostics. By processing sensor data locally, edge computing improves fault detection speed, reduces network dependency, and enhances data privacy. As edge computing evolves, integrating algorithms like LWA will drive advancements in real-time fault detection and autonomous driving technologies.

When compared to CNN-based models, which require large datasets for training, our model demonstrates a significant advantage in terms of computational efficiency. The lightweight design of the LWA ensures that the model can run on embedded automotive systems, making it suitable for real-time applications, unlike CNNs, which are computationally intensive.

References

[1] Chakrapani, G., & Sugumaran, V. " ," Engineering Applications of Artificial Intelligence, vol. 117, pp. 105522, 2023. https://doi.org/10.1016/j.engappai.2022.105522.

[2] Hossain, M. N., Rahman, M. M., & Ramasamy, D. "Artificial Intelligence-Driven Vehicle Fault Diagnosis to Revolutionize Automotive Maintenance: A Review," CMES-Computer Modeling in Engineering and Sciences, vol. 141, no. 2, pp. 951-996, 2024. https://doi.org/10.32604/cmes.2024.056022.

[3] Hu, C., Zhang, Z., Li, C., Leng, M., Wang, Z., Wan, X., & Chen, C. "A state of the art in digital twin for intelligent fault diagnosis," Advanced Engineering Informatics, vol. 63, pp. 102963, 2025. https://doi.org/10.1016/j.aei.2024.102963.

[4] Jegadeeshwaran, R., & Sugumaran, V. "Fault diagnosis of automobile hydraulic brake system using statistic features and support vector machines," Mechanical Systems and Signal Processing, vol. 52-53, pp. 436-446, 2015. https://doi.org/10.1016/j.ymssp.2014.08.007.

[5] Govindasamy, R., Nagarajan, S. K., Muthu, J. R., & Ramkumar, M. "Residual multiscale attention based modulated convolutional neural network for radio link failure prediction in 5G,". Ad Hoc Networks, vol. 166, pp. 103679, 2025. https://doi.org/10.1016/j.adhoc.2024.103679.

[6] Chakraborty, S., Saha, A. K., Ezugwu, A. E., Chakraborty, R., & Saha, A. "Horizontal crossover and co-operative hunting-based Whale Optimization Algorithm for feature selection," Knowledge-Based Systems, vol. 282, pp. 111108, 2023. https://doi.org/10.1016/j.knosys.2023.111108.

[7] Chandrashekar, C., Krishnadoss, P., Poornachary, V. K., & Ananthakrishnan, B. "MCWOA Scheduler: Modified Chimp-Whale Optimization Algorithm for Task Scheduling in Cloud Computing," Computers, Materials and Continua, vol. 78, no. 2, pp. 2593-2616, 2024. https://doi.org/10.32604/cmc.2024.046304.

[8] Dao, T.-K., & Nguyen, T.-T. "An Optimal Node Localization in WSN Based on Siege Whale Optimization Algorithm," CMES-Computer Modeling in Engineering and Sciences, vol. 138, no. 3, pp. 2201-2237, 2023. https://doi.org/10.32604/cmes.2023.029880.

[9] Gupta, B. B., Gaurav, A., Attar, R. W., Arya, V., Alhomoud, A., & Chui, K. T. "Optimized Phishing Detection with Recurrent Neural Network and

Whale Optimizer Algorithm," Computers, Materials and Continua, vol. 80, no. 3, pp. 4895-4916, 2024. https://doi.org/10.32604/cmc.2024.050815.

[10] Hasan, M. W. "Building an IoT temperature and humidity forecasting model based on long short-term memory (LSTM) with improved whale optimization algorithm," Memories-Materials, Devices, Circuits and Systems, vol. 6, pp. 100086, 2023. https://doi.org/10.1016/j.memori.2023.100086.

[11] Hosseinzadeh, M., Tanveer, J., Alanazi, F., Aurangzeb, K., Yousefpoor, M. S., Yousefpoor, E., Darwesh, A., Lee, S.-W., & Rahmani, A. M. "An intelligent clustering scheme based on whale optimization algorithm in flying ad hoc networks," Vehicular Communications, vol. 49, pp. 100805, 2024. https://doi.org/10.1016/j.vehcom.2024.100805.

[12] Kumar, N., Singh, K., & Lloret, J. "WAOA: A hybrid whale-ant optimization algorithm for energy-efficient routing in wireless sensor networks," Computer Networks, vol. 254, pp. 110845, 2024. https://doi.org/10.1016/j.comnet.2024.110845.

[13] Li, M.-W., Xu, R.-Z., Yang, Z.-Y., Yeh, Y.-H., & Hong, W.-C. "Optimizing berth-crane allocation considering tidal effects using chaotic quantum whale optimization algorithm," Applied Soft Computing, vol. 162, pp. 111811, 2024. https://doi.org/10.1016/j.asoc.2024.111811.

[14] Miao, F., Wu, Y., Yan, G., & Si, X. "A memory interaction quadratic interpolation whale optimization algorithm based on reverse information correction for high-dimensional feature selection," Applied Soft Computing, vol. 164, pp. 111979, 2024. https://doi.org/10.1016/j.asoc.2024.111979.

[15] Miao, F., Wu, Y., Yan, G., & Si, X. "Dynamic multi-swarm whale optimization algorithm based on elite tuning for high-dimensional feature selection classification problems," Applied Soft Computing, vol. 169, pp. 112634, 2025. https://doi.org/10.1016/j.asoc.2024.112634.

[16] Sahayaraj, J. M., Gunasekaran, K., Verma, S. K., & Dhurgadevi, M. "Energy efficient clustering and sink mobility protocol using Improved Dingo and Boosted Beluga Whale Optimization Algorithm for extending network lifetime in WSNs," Sustainable Computing: Informatics and Systems, vol. 43, pp. 101008, 2024. https://doi.org/10.1016/j.suscom.2024.101008.

[17] Guo, B., Qiu, S., Zhang, P., & Tang, X. "Mural Anomaly Region Detection Algorithm Based on Hyperspectral Multiscale Residual Attention Network,". Computers, Materials and Continua, vol. 81, no. 1, pp. 1809–1833, 2024. https://doi.org/10.32604/cmc.2024.056706.

[18] Xue, Z., Yi, X., Feng, W., Kong, L., & Wu, M. "Prediction and mapping of soil thickness in alpine canyon regions based on whale optimization algorithm optimized random forest: A case study of Baihetan Reservoir area in China," Computers & Geosciences, vol. 191, pp. 105667, 2024. https://doi.org/10.1016/j.cageo.2024.105667.

[19] Yang, D., Zhou, C., Wei, X., Chen, Z., & Zhang, Z. "Multi-Strategy Assisted Multi-Objective Whale Optimization Algorithm for Feature Selection" CMES-Computer Modeling in Engineering and Sciences, vol. 140, no. 2, pp. 1563-1593, 2024. https://doi.org/10.32604/cmes.2024.048049.

[20] Khan, T. M., Naqvi, S. S., & Meijering, E. "ESDMR-Net: A lightweight network with expand-squeeze and dual multiscale residual connections for medical image segmentation,". Engineering Applications of Artificial Intelligence, vol. 133, pp. 107995, 2024. https://doi.org/10.1016/j.engappai.2024.107995.

[21] Zhou, H., Feng, R., Peng, Y., Jin, D., Li, X., Shou, D., Li, G., & Wang, L. "Integration of multiscale fusion of residual neural network with 2-D gramian angular fields for lower limb movement recognition based on multi-channel sEMG signals,". Biomedical Signal Processing and Control, vol. 99, pp. 106807, 2025. https://doi.org/10.2139/ssrn.4801264.

[22] Jiang, F., Kuang, Y., Li, T., Zhang, S., Wu, Z., Feng, K., & Li, W. "Towards Enhanced Interpretability: A Mechanism-Driven domain adaptation model for bearing fault diagnosis across operating conditions," Mechanical Systems and Signal Processing, vol. 225, pp. 112244, 2025. https://doi.org/10.1016/j.ymssp.2024.112244.

[23] Jiang, X., Song, Q., Wang, Q., Zhang, W., Ding, C., & Zhu, Z. "Spectral boundary detecting model: A promising tool for adaptive mode extraction and machinery fault diagnosis," Advanced Engineering Informatics, vol. 61, pp. 102494, 2024. https://doi.org/10.1016/j.aei.2024.102494.

[24] Li, C., Lu, P., & Chen, G. "VNCCD: A gearbox fault diagnosis technique under nonstationary conditions via virtual decoupled transfer path," Mechanical Systems and Signal Processing, vol. 221, pp. 111741, 2024. https://doi.org/10.1016/j.ymssp.2024.111741.

[25] Milfont, L. D., Ferreira, G. T. de C., & Giesbrecht, M. "Fault diagnosis in electric machines and propellers for electrical propulsion aircraft: A review," Engineering Applications of Artificial Intelligence, vol. 139, pp. 109577, 2025. https://doi.org/10.2139/ssrn.4823375.

[26] Shandhoosh, V., S, N. V., Chakrapani, G., Sugumaran, V., Ramteke, S. M., & Marian, M. "Intelligent fault diagnosis for tribo-mechanical systems by machine learning: Multi-feature extraction and ensemble voting methods," Knowledge-Based Systems, vol. 305, pp. 112694, 2024. https://doi.org/10.1016/j.knosys.2024.112694.

[27] Tang, S., Ma, J., Yan, Z., Zhu, Y., & Khoo, B. C. "Deep transfer learning strategy in intelligent fault diagnosis of rotating machinery," Engineering Applications of Artificial Intelligence, vol. 134, pp. 108678, 2024. https://doi.org/10.1016/j.engappai.2024.108678.

[28] Wang, L., Li, Y., Liu, J., Peng, J., Zhang, Q., & Fu, W. "Research on fault diagnosis of industrial robots based on generative adversarial network," Physical Communication, vol. 64, pp. 102355, 2024. https://doi.org/10.1016/j.phycom.2024.102355.

Appendix

Reproducibility and implementation details

Input: 32-dimensional feature vector
Conv1: 1×1 convolution, 64 filters, ReLU activation
Conv2: 3×3 dilated convolution (dilation=2), 64 filters, ReLU
MaxPooling: 2×2
Global Average Pooling
Fully Connected Layer: 64 → 6 (fault categories)
Softmax Output