PLOS One

RESEARCH ARTICLE

A fusion sparse learning algorithm for fault identification of rolling bearings

Yefeng Liu[1,2], **Jingjing Liu**[1,3]*, **Yanwei Ma**[1,4], **Shuai Wang**[5], **Qichun Zhang**[6]

1 Liaoning Key Laboratory of Information Physics Fusion and Intelligent Manufacturing for CNC Machine, Shenyang Institute of Technology, Fushun, Liaoning, China, **2** School of Automation and Electrical Engineering, Linyi University, Linyi, Shandong, China, **3** Department of Basic Courses, Shenyang Institute of Technology, Fushun, Liaoning, China, **4** School of Mechanical Engineering and Automation, Shenyang Institute of Technology, Fushun, Liaoning, China, **5** School of Automation and Electrical Engineering, Shenyang Ligong University, Shenyang, Liaoning, China, **6** School of Creative and Digital Industries, Buckinghamshire New University, High Wycombe, United Kingdom

* liujingjing@situ.edu.cn

OPEN ACCESS

Citation: Liu Y, Liu J, Ma Y, Wang S, Zhang Q (2026) A fusion sparse learning algorithm for fault identification of rolling bearings. PLoS One 21(1): e0339859. https://doi.org/10.1371/journal.pone.0339859

Editor: Kang-Di Lu, Donghua University, CHINA

Received: March 1, 2025

Accepted: December 11, 2025

Published: January 5, 2026

Data availability statement: This study analyzes third-party data from the Case Western Reserve University (CWRU) Bearing Data Center, which we do not own or have legal authority to redistribute.

Abstract

A key part of CNC machine tools is the rolling bearing, and thus, it is vital to employ a data-driven approach for fault diagnosis. This paper proposes a two-stage fusion sparse learning algorithm for fault data processing that can identify and diagnose the fault types of rolling bearings based on sensor measurement data. During the feature extraction phase, temporal features of sequential data within the big data are extracted using a Long Short - Term Memory (LSTM) network. Moreover, the classification learning stage contains a new sparse learning algorithm, which applies $L_{1/2}$ regularization on stochastic configuration networks (SCN). The iterative learning formula combines the alternating direction method of multipliers (ADMM) with the analysis of the quadratic equations theory. Simultaneously, the model's inequality supervision mechanism is updated based on convergence analysis. This developed algorithm incorporates the benefits of LSTM in extracting temporal data characteristics, along with the sparsity, ease of convergence, and lightweight nature of SCN. Consequently, it mitigates the shortcomings of deep models in end-to-end applications, particularly in terms of interpretability and structural redundancy, thus making it suitable for deployment on edge devices. Finally, a fusion sparse learning model (LSTM-$L_{1/2}$-SCN) is introduced based on the two-stage learning algorithm for rolling bearing fault diagnosis. In the experiments on the benchmark dataset, the optimal sparsity degree of this algorithm for the Sparse Coding Network (SCN) reached 76.66%, which was 30% higher than that of the Pooling-based Sparse Coding Network (PSCN). Moreover, in the experiments based on the dataset of Case Western Reserve University (CWRU), the optimal test classification accuracy achieved was 97.51%, and the optimal sparsity degree for SCN reached 29.39%. These results

The dataset used for our analysis is publicly available without restrictions at: https://engineering.case.edu/bearingdatacenter/download-data-file. All researchers may access the data identically to the authors via this link. The authors confirm they had no special access privileges unavailable to others.

Funding: This research was funded by Science and Technology Plan Project of Liaoning Province (2023JH2/101700066), National Science Foundation of China under Grants (62073226), Basic Scientific research project of Liaoning province Department of Education (JYTMS20231179), State Key Laboratory of Synthetical Automation for Process Industries Program (2023-kfkt-03, SAPI-2024-KFKT-05). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

verify that the proposed algorithm exhibits sparsity, demonstrates effectiveness, and is capable of identifying faults in rolling bearings.

Introduction

Rolling bearings are essential to CNC machine tools, significantly affecting CNC's regular operation. Specifically, the outer ring, inner ring, or rolling part of the rolling bearing is most prone to wear or deformation under high-load operation, affecting the entire production process. Therefore, fault prediction and diagnosis of rolling bearings are significant. Due to the swift advancement of deep learning, data-driven fault diagnosis of rolling bearings has gained increasing popularity. In such strategies, data acquisition is realized by sensors and measured by signal processing methods. Vibration signal analysis is one of the most studied sensing methods at present.

Traditional vibration signal analysis methods rely on manual feature extraction and are difficult to adapt to complex working conditions, such as Fourier transform and wavelet decomposition, Vector Local Characteristic-Scale Decomposition (Vector LCD) [2], fuzzy signal feature fusion technology [3], Principal Component Analysis (PCA), and digital twin and transfer learning [4,5]. Although deep learning models (such as CNN and LSTM) have achieved automatic feature extraction through end-to-end learning, they still face two major challenges in practical industrial applications: High model complexity leads to difficult deployment (for example, the parameter count of ResNet-50 reaches 23M), and it is challenging to systematically analyze the convergence of the model.

The current research on bearing fault diagnosis mainly falls into three categories of methods: Deep learning methods: numerous models have garnered extensive application [6–8], with the main models including CNN, Deep Belief Network (DBN), Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU) [9], Long Short-Term Memory (LSTM) and Resnet [10]. In [11], the authors demonstrated that combining multi-scale CNN and LSTM models can efficiently diagnose bearing faults. CNN can also be combined with a multi-layer perceptron [12] or multi-task model [13]. CNN algorithms mentioned in [14–19] have also been successfully applied to the field of rolling bearings. Concurrently, recent advancements have led to more efficient architectures such as EPyNet, an energy-efficient 1D-CNN architecture, which achieves significant energy reduction and high accuracy on multiple audio emotion recognition datasets while being compatible with CPU and resource-constrained edge devices [20]. In recent years, combining deep learning with attention mechanisms has yielded promising results, with representative methods being Attentive dense CNN [21], Attention-temporal convolutional neural networks (ATCN), Attention-LSTM, Convolutional Bi-Directional LSTM (CBLSTM) [22], 1DCNN-LSTM [23], TCN-BiLSTM, and Attention TCN-BiLSTM [24].These models proposed attained an accuracy surpassing 90% on the CWRU dataset, but it requires GPU acceleration and cannot explain the decision-making basis. [25] combines CNN with the self-attention of Transformer to achieve efficient computing on mobile devices. In the ImageNet classification task, the model with 0.701M parameters was superior to the pure Transformer scheme.

Lightweight models: SVM, KNN, and SCN [26] are computationally efficient, with SCN converging under an inequality supervision mechanism. However, SCN's sparsity and generalization capabilities require further improvement to facilitate lightweight deployment. Regularization techniques, including L_1, L_2, and smooth L_1 regularization, have been applied to enhance these aspects [27–29]. Among them, $\mathrm{L}_{1/2}$ regularization is particularly effective in generating sparser solutions, offering a more accurate model representation while preserving sparsity [30]. The sparsity and generalization performance of L_1/L_2-regularized SCN is insufficient. Further optimization of SCN's sparsity and generalization is still desirable.

Hybrid architectures combining deep and shallow models: Hybrid models can combine the respective advantages of deep and shallow models in feature extraction and achieving model lightweighting and sparsity. Unfortunately, there are not many cases of fusing deep and shallow models for phased prediction at present. The ones that have been proposed so far include: LSTM-SVM, which uses LSTM for signal prediction followed by SVM for mechanical state diagnosis [31], and the CNN-LSTM-SVM, which extracts signal features via CNN and LSTM before SVM-based fault classification [32]. The average accuracy rate for fault classification achieved by these models exceeds 95.92% on the CWRU dataset.

Current hybrid models are constrained by two main issues: CNN-based approaches are inadequate for representing time-varying fault characteristics like impact periodicity, and the shallow classifiers used lacks the global approximation capabilities like SCN and is not sufficiently sparse.

In response to the above problems, this paper proposes a novel diagnostic framework that integrates LSTM and $\mathrm{L}_{1/2}$ regularized SCN. The main contributions include:

1) An $\mathrm{L}_{1/2}$ regularization solution algorithm based on the roots of cubic equations is proposed. Theoretically, it is proved that it has a better sparse error bound than L_1 regularization. Construct an incremental supervision mechanism to guarantee that the model converges to a certain extent and simultaneously enhances its feature selection ability. 2) Design a hierarchical feature processing architecture: The LSTM layer extracts temporal features, and the $\mathrm{L}_{1/2}$-SCN layer conducts sparse classification. 3) On the CWRU dataset, experimental evaluations demonstrate that the proposed model achieves a 0.64% improvement in average classification accuracy and attains 23.44% model sparsity when compared with state-of-the-art benchmarks including TCN-LSTM, TCN-BiLSTM, ResNet architectures, and other representative methods.

The remainder of this article is organized as follows. The second part introduces the preliminary knowledge about LSTM and SCN. The third part proposes a sparse learning algorithm based on $\mathrm{L}_{1/2}$ regularization and then provides the fusion sparse learning algorithm. The fourth part conducts some numerical experiments to verify the effectiveness of the proposed algorithm. Finally, the fifth part summarizes this paper.

Preliminaries

Feature extraction method based on LSTM

The state of the system at a certain moment is determined by the combined influence of its past state and the current input. Since the system's state evolves over time based on these factors, the signals processed by the system are inherently time-dependent. The core design objective of LSTM is to handle sequential data. It can autonomously learn to remember long-term information, forget irrelevant information, and focus on the current input through the forget gate, input gate, and output gate, which makes it well-suited for handling vibration signals with long-term trends and periodic patterns. Meanwhile, LSTM offers a low-attenuation path for gradient backpropagation through cell states and gating mechanisms, thereby effectively alleviating the problem of vanishing gradients. Compared with CNN, which is better at extracting local regional features from signals, LSTM has more advantages in extracting features from sequential data. Fig 1 illustrates the schematic representation of LSTM's architecture, where f_t, i_t, and o_t represent the forget gate, input gate, and output gate, respectively. Besides, c_t and h_t represent the state of the cell and hidden layer at time t, σ and tanh are activation functions. LSTM, through its gated architecture, effectively captures both short-term and long-term (h and c) dependencies in sequential data, making it particularly suitable for tasks like natural language processing and time series

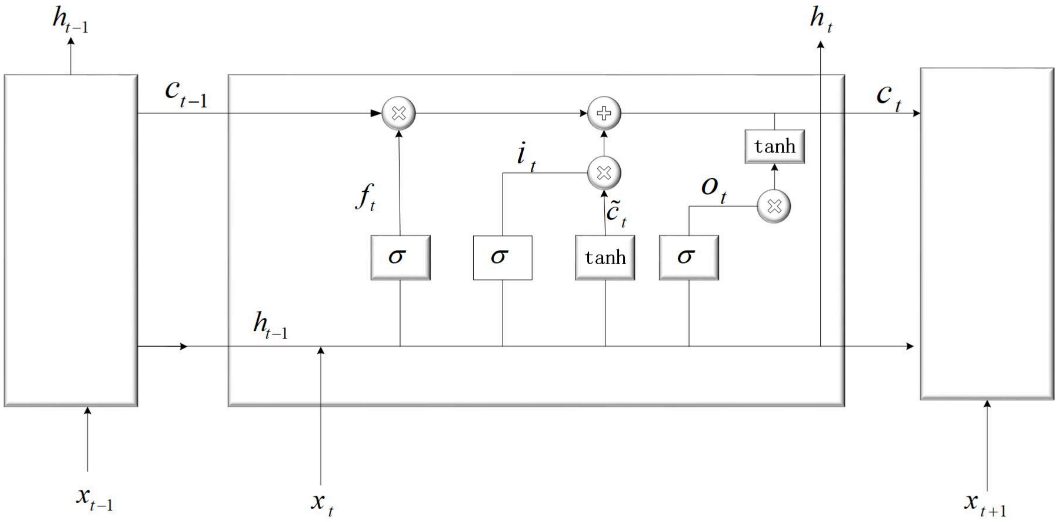


Fig 1. **The structure of LSTM.**

https://doi.org/10.1371/journal.pone.0339859.g001

forecasting. Specifically, the gating mechanism within LSTM enables data to be added, discarded, and stored within the cell. The forgetting gate f_t processes the forgotten information from c_{t-1} and preserves the stored data in the current state. The input gate captures the current information, which is then used to compute a candidate cell state c_t combined with the previous cell state c_{t-1} to generate the new cell state c_t. Meanwhile, the output gate o_t determines what part of the cell state c_t is used to create the hidden state h_t for the current time step. The final output represents a comprehensive representation of the current states, and the data flow within LSTM is calculated as follows:

$$f(t) = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\hat{c}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

$$c_t = f_t c_{t-1} + i_t c_t \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \cdot tanh(c_t) \tag{6}$$

where W_i, W_f, W_c and W_o represent the input gate, forget gate, current status, and output gate weights, respectively, and b_i, b_f, b_c and b_o represent the corresponding bias. To improve the learning performance and obtain more specific data features, LSTM is used to extract the time features, and the output h of the hidden layer is used as the data features. The output of LSTM reflects the relevant historical information. Due to its superiority in processing time series data, this paper does not employ a complex deep model for end-to-end fault diagnosis processing. LSTM is selected for the feature extraction stage.

Principles of stochastic configuration networks

Let $X=(x^1,x^2,...,x^N)^T$ be the input data, where $x^i=(x^i_1,x^i_2,...,x^i_d)$. $T=(t^1,t^2,...,t^N)^T$ are the corresponding output data , where $t^i=(t^i_1,t^i_2,...,t^i_m)$, N signifies the quantity of samples, d denotes the dimensionality of input features, and m represents the count of output features. The structure of SCN with L hidden nodes is depicted in Fig 2.

Let the weights and biases between the input and hidden layer be $W=(w_1,w_2,...,w_L)$, $b=(b_1,b_2,...,b_L)$, where $w_l=(w_{l1},w_{l2},...,w_{ld})^T$, $l=1,2,...,L$, $b_l\in R$. Then, the output of the L-th hidden node and outputs of all hidden nodes are formulated in (7) and (8).

$$h_l=g(Xw_l+b_l) \tag{7}$$

$$H=(h_1,h_2,...,h_L)=g(XW+b) \tag{8}$$

where $X\in\mathbb{R}^{N\times d}$, $H\in\mathbb{R}^{N\times L}$, $W\in\mathbb{R}^{d\times L}$, g is the activation function. The weight between the hidden and output layer is $\beta=(\beta_1,\beta_2,...,\beta_L)^T$, where $\beta\in\mathbb{R}^{L\times m}$, $\beta_l=(\beta_{l1},\beta_{l2},...,\beta_{lm})$, the output of SCN is

$$y(L)=(y_1(L),y_2(L),...,y_m(L))=H\beta=\sum_{l=1}^{L}h_l\beta_l \tag{9}$$

the error is

$$e_L=T-H\beta=(e_{L1},e_{L2},...,e_{Lm}) \tag{10}$$

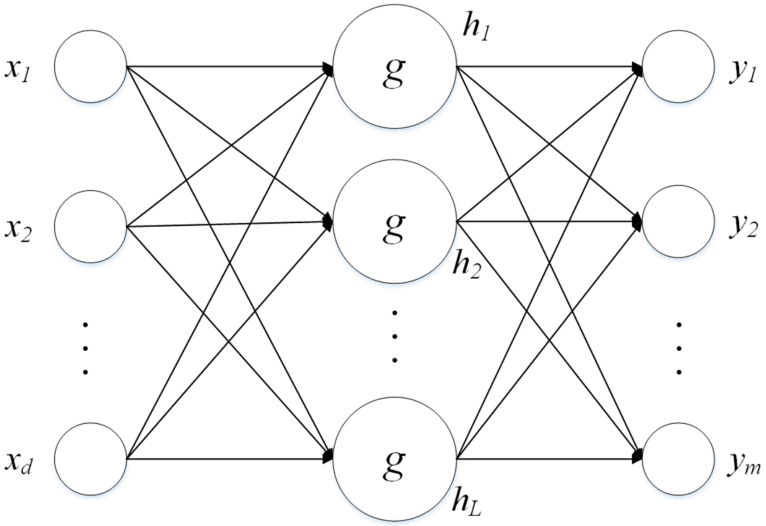


Fig 2. The structure of SCN with L hidden nodes.

https://doi.org/10.1371/journal.pone.0339859.g002

where $T \in \mathbb{R}^{N \times m}$, then the equation $e_{L+1} = e_L - h_{L+1}\beta_{L+1}$ can be get based on the equation $y(L+1) = y(L) + h_{L+1}\beta_{L+1}$. The construction process of the model begins with its initialization, setting $y(0) = 0$. Subsequently, e_0 is calculated as $T - y(0) = T$. When the L-th node is generated, the choice of w_L and b_L follows the inequality supervision mechanism.

$$\sum_{q=1}^{m} \langle e_{L-1,q}, h_L \rangle^2 \geq b_g^2(1 - r - \mu_L)||e_{L-1}||^2 \tag{11}$$

where $e_{L-1,q}$ represents the q-th dimension error after the L-1 hidden node has been configured, b_g is the upper bound of the activation function, r is a constant close to 1, and the real number sequence $\{\mu_L\}$ is satisfied $\lim_{L \to +\infty} \mu_L = 0$. The inequality constraint in Eq (11) forms the theoretical foundation for SCN stability by guaranteeing monotonic error reduction during incremental construction. This inequality supervision is essential because: (i) it ensures each new hidden node decreases the residual error to guarantee the convergence of the network, preventing network overgrowth; (ii) the parameters r and μ create a contraction mapping that guarantees convergence. Without this constraint, random node addition could cause oscillating or divergent training behavior. β can be determined by the global least square method by (12).

$$\beta^* = \underset{\beta}{argmin} ||H\beta - T||^2 = H^{\dagger} T \tag{12}$$

When the first node has been configured (w_1,b_1, and β_1 are determined), the above steps are repeated to gradually increase the nodes and guide the predetermined maximum number or accuracy.

The fusion sparse learning algorithm

The sparse learning algorithm of $L_{1/2}$-SCN

The unregularized SCN employs least squares for weight estimation, often resulting in numerical instability and overfitting. While L_1 regularization improves sparsity and reduces model complexity. Theoretical analysis demonstrates that $L_{1/2}$ regularization possesses stronger sparsity-inducing properties than L_1 regularization [30], $L_{1/2}$ regularization strikes an optimal balance between L_0 sparsity and L_1 tractability, and its non-convex formulation better approximates L_0's sparsity while remaining computationally feasible. Meanwhile, in practical scenarios with limited samples, its adaptive thresholding mechanism provides superior noise-feature discrimination by selectively preserving weak but diagnostically significant fault characteristics.

$L_{1/2}$ regularization is an effective sparsity method that improves the error function of SCN, specifically by adding the $L_{1/2}$ regularization term to the objective function, as presented in (13). Here, λ is the regularization coefficient.

$$\min_{\beta} : \frac{1}{2}||H\beta - T||^2 + \lambda||\beta||_{1/2}^{1/2} \tag{13}$$

The Admm algorithm is used to solve the $L_{1/2}$ regularization problem, and the specific methods are described below. Construct the optimization problem:

$$min : f(x) + g(\beta) = \frac{1}{2}||Hx - T||^2 + \lambda||\beta||_{1/2}^{1/2} \tag{14}$$

$$s.t. \quad x - \beta = 0 \tag{15}$$

Let $\mu_1 = \frac{\mu}{\rho}$, the original problem is equivalent to solving the following problem

$$x^{k+1} = \underset{x}{argmin}(f(x) + \frac{\rho}{2}\|x - \beta^k + \mu_1^k\|_2^2) \tag{16}$$

$$\beta^{k+1} = \underset{\beta}{argmin}(g(\beta) + \frac{\rho}{2}\|x^{k+1} - \beta + \mu_1^k\|_2^2) \tag{17}$$

$$\mu_1^{k+1} = \mu_1^k + x^{k+1} - \beta^{k+1} \tag{18}$$

solve for (16) and the following equation can be get

$$x^{k+1} = (H^T H + \rho I)^{-1}(H^T T + \rho(\beta^k - \mu_1^k)) \tag{19}$$

Take the derivative of β in formula (17), search the stagnation point, get the equation (20)

$$\rho\beta + \frac{\lambda}{2} \cdot \frac{sign(\beta)}{\sqrt{(\beta)}} - \rho(x^{k+1} + \mu_1^k) = 0 \tag{20}$$

a. If $\beta > 0$, let $t = \sqrt{(|\beta|)}$, then formula (20) is converted to (21)

$$t^3 - (x^{k+1} + \mu_1^k)t + \frac{\lambda}{2\rho} = 0 \tag{21}$$

let $m = x^{k+1} + \mu_1^k$, $n = \frac{\lambda}{2\rho}$, it can be seen from the cubic equation and the graph form that when the discriminant $\Delta = (\frac{n}{2})^2 - (\frac{m}{3})^3 < 0$, namely $(\frac{m}{3})^3 > (\frac{n}{2})^2$, the equation has three unequal real roots. According to Cartan's formula, the roots of the equation are one negative and two positive, and the largest positive root is the minimum point of (17), as shown in (22) and (23). For the $L_{1/2}$ regularization problem, Xu et al. [30] proved that the objective function is unimodal on the positive real axis, with its unique critical point (the maximum root of the cubic equation) guaranteed to correspond to a local minimum, as verified through second-order convexity analysis.

$$t = 2\sqrt{(|m|/3)} \cdot cos(\frac{\pi}{3} - \frac{\phi}{3}) \tag{22}$$

$$\phi = arccos(\frac{n}{2} \cdot (\frac{|m|}{3})^{\frac{-3}{2}}) \tag{23}$$

therefore

$$\beta = \frac{4|m|}{3} \cdot cos^2(\frac{\pi}{3} - \frac{\phi}{3}) \tag{24}$$

b. If $\beta < 0$, let $t = \sqrt{(|\beta|)}$, then formula (20) is converted to (25)

$$t^3 + (x^{k+1} + \mu_1^k)t + \frac{\lambda}{2\rho} = 0 \tag{25}$$

In a similar way, when $(\frac{m}{3})^3 < -(\frac{n}{2})^2$

$$\beta = -\frac{4|m|}{3} \cdot cos^2(\frac{\pi}{3} - \frac{\phi}{3}) \tag{26}$$

Then the optimal solution of the objective function is

$$\beta^{k+1} = \begin{cases} \frac{4|m|}{3} \cdot cos^2(\frac{\pi}{3} - \frac{\phi}{3}), & (\frac{m}{3})^3 > (\frac{n}{2})^2 \\ 0, & others \\ -\frac{4|m|}{3} \cdot cos^2(\frac{\pi}{3} - \frac{\phi}{3}), & (\frac{m}{3})^3 < -(\frac{n}{2})^2 \end{cases} \tag{27}$$

The update formula of μ_1 can be equivalently converted from (18) to (28).

$$\mu_1^{k+1} = \mu_1^k + x^{k+1} - \beta_q^{k+1} \tag{28}$$

In summary, the sparse learning algorithm of $L_{1/2}$ regularized SCN is given by iterative solution according to formulas (19), (27) and (28).

Inequality supervision mechanism for $L_{1/2}$-SCN

Analysis the objective function

$$J = \frac{1}{2}||H\beta - T||^2 + \lambda||\beta||_{1/2}^{1/2} \tag{29}$$

$$= \frac{1}{2}||e_{L-1} - g_L\beta_L||^2 + \lambda\sum(|\beta|^{1/2}) \tag{30}$$

$$\frac{\partial J}{\partial \beta_L} = -g_L(e_{L-1} - g_L\beta_L) + \frac{\lambda}{2} \cdot \frac{sign(\beta_L)}{\sqrt{(}|\beta_L|)} = 0 \tag{31}$$

in the same way as the solution for (20), let $ieql = (\frac{\langle e_{L-1}, g_L\rangle}{3g_L^2})^3$,$ieqr = 2(\frac{\lambda}{4g_L^2})^2$,we can get

$$\beta_L = \begin{cases} \frac{4}{3}|\frac{\langle e_{L-1}, g_L\rangle}{g_L^2}| \cdot cos^2(\frac{\pi}{3} - \frac{\phi}{3}), & ieql > ieqr \\ 0, & |ieql| \leq ieqr \\ -\frac{4}{3}|\frac{\langle e_{L-1}, g_L\rangle}{g_L^2}| \cdot cos^2(\frac{\pi}{3} - \frac{\phi}{3}), & ieql < -ieqr \end{cases} \tag{32}$$

In SCN, the choice of w and b need to satisfy the inequality $||e_{L-1}||^2 - ||e_L||^2 \geq (1 - r - \mu)||e_{L-1}||^2$, namely

$$2\langle e_{L-1}, g_L\rangle\beta_L - \beta_L^2 g_L^2 \geq (1 - r - \mu)||e_{L-1}||^2 \tag{33}$$

Let

$$v_1 = \frac{|\langle e_{L-1}, g_L\rangle|}{g_L^2} \cdot cos^2(\frac{\pi}{3} - \frac{\phi}{3}) \tag{34}$$

$$v_2 = \frac{8}{3}\langle e_{L-1}, g_L\rangle \tag{35}$$

$$v_3 = \frac{16}{9}|\langle e_{L-1}, g_L\rangle| \cdot cos^2(\frac{\pi}{3} - \frac{\phi}{3}) \tag{36}$$

An inequality supervision mechanism for L$_{1/2}$-SCN is obtained by substituting the expression for β_L into (33), as shown in (37) and (38). The following conclusions can be drawn:

if

$$(\frac{\langle e_{L-1}, g_L\rangle}{3g_L^2})^3 > 2(\frac{\lambda}{4g_L^2})^2) \tag{37}$$

the inequality supervision mechanism is:

$$v_1(v_2 - v_3) \geq (1 - r - \mu_L)||e_{L-1}||^2, \tag{38}$$

and if

$$(\frac{\langle e_{L-1}, g_L\rangle}{3g_L^2})^3 < -2(\frac{\lambda}{4g_L^2})^2) \tag{39}$$

the inequality supervision mechanism is:

$$-v_1(v_2 + v_3) \geq (1 - r - \mu_L)||e_{L-1}||^2, \tag{40}$$

Therefore, new hidden nodes are incrementally added when either condition (37) or (39) is satisfied, strictly following the inequality constraints specified in (38) or (40) respectively. If neither condition is met, according to (32), the corresponding weight is set to zero. It is noteworthy that the inequality supervision mechanism proposed above enables the model to converge to a certain degree. Nevertheless, during the sparse - processing procedure, β is set to zero without fulfilling the inequality constraints. Formula (32), (37)–(40) reveal the contradictions inherent in these two aspects. Consequently, in practical applications, it is imperative to strike a balance between sparsity and model accuracy. The $L_{1/2}$-SCN proposed in this section theoretically analyzes its own convergence and updates the original inequality supervision mechanism. This update allows the algorithm to offer a sparser model representation, which is advantageous for actual fault identification and classification tasks. In the entire fault diagnosis process, the algorithm can take over the feature extraction task from the previous stage to facilitate fault type identification. The algorithm flow of $L_{1/2}$-SCN is shown in Algorithm 1.

Fusion sparse learning algorithm

Massive information, inherent noise, temporal dependencies, and pronounced periodicity typically govern signal data. Utilizing a singular model for learning may hinder the thorough examination of the underlying patterns within the data. To confront the intricacies arising from voluminous datasets and vague features, this study employs LSTM to extract temporal features. Subsequently, these features are input for sparse learning via the L$_{1/2}$-SCN model, enhancing its performance and resulting in a sparse structural representation. Fig 3 depicts the architecture of the fusion model, while the detailed algorithmic steps are outlined in Algorithm 2.

Standardized Feature Fusion Pipeline is as follows:

(1) Temporal Feature Extraction The original input data $X \in \mathbb{R}^{N\times d}$ undergoes feature extraction through a single-layer LSTM network configured with hidden units:

$$H\_LSTM_t = LSTM(X; W_h, U_h, b_h), \quad H\_LSTM_t \in \mathbb{R}^{N\times h} \tag{41}$$

Algorithm 1 Incremental node addition with adaptive regularization.

```
Require:
    $\mathbf{X} \in \mathbb{R}^{N \times d}$                    ▷ Input data matrix (N samples × d features)
    $\mathbf{T} \in \mathbb{R}^{N \times m}$                    ▷ Target matrix
    $\xi = 10^{-2}$                    ▷ Residual error tolerance (convergence threshold)
    $L_{\max} = 500$                    ▷ Maximum hidden nodes
    $T_{\max} = 30$                    ▷ Max attempts per regularization parameter
Ensure:
    $\beta = [\beta_1, \beta_2, ..., \beta_L]$,                    ▷ Output weights
    $w^* = [w_1^*, w_2^*, ..., w_L^*]$,  $b^* = [b_1^*, b_2^*, ..., b_L^*]$                    ▷ Optimal node parameters
1:  Initialize: $e_0 \leftarrow \mathbf{T}, r \leftarrow 0.9$                    ▷ Initialization
2:  Set regularization grid: $\gamma \leftarrow \{1, 5, 10, 20, 30\}$                    ▷ $\Lambda$ search range
3:  while $L \le L_{\max}$ and $\|e_L\|_2 > \xi$ do
4:      for $\Lambda \in \gamma$ do                    ▷ Adaptive node generation
5:          for $k = 1$ to $T_{\max}$ do
6:              Sample $w_L \sim \mathcal{U}(-\Lambda, \Lambda)$,  $b_L \sim \mathcal{U}(-\Lambda, \Lambda)$                    ▷ Random projection
7:              if Inequality (32) satisfied then
8:                  $W \leftarrow W \cup \{w_L\}$,                    ▷ Archive valid nodes
9:              end if
10:         end for
11:         if $W \neq \varnothing$ then
12:             Select $(w_L^*, b_L^*)$ maximizing formula (33)                    ▷ Node selection
13:             break (textbfgoto (15))
14:         else
15:             Adjust $r \leftarrow r + \tau$, $\tau \sim \mathcal{U}(0, 1 - r)$,Return 4                    ▷ Relax supervision
16:         end if
17:     end for
18:     Compute $\beta^*$ via Eq (19), (27), (28)                    ▷ Least squares solution
19:     Update residual: $e_L \leftarrow e_{L-1} - \beta_L h_L^*$, $h_L^* = [g_1^*, g_2^*, ..., g_L^*]$                    ▷ Error correction
20:     $L \leftarrow L + 1$
21: end while
```

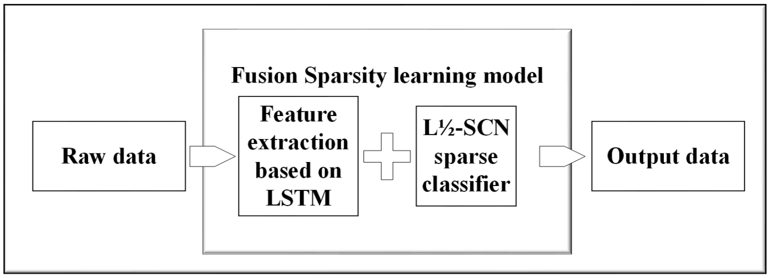

Fusion Sparsity learning model
Raw data
Feature extraction based on LSTM
L½-SCN sparse classifier
Output data

Fig 3. The fault identification method utilizing fusion sparse learning model.

https://doi.org/10.1371/journal.pone.0339859.g003

where W_h, U_h and b_h denote the input weights, recurrent weights, and bias terms, respectively.

(2) Standardize the output H_LSTM_t of the LSTM layer. (The numerical range of LSTM hidden states is influenced by both the input data's physical dimensions and the activation function, potentially resulting in magnitude variations across different samples. Therefore, the data needs to be normalized)

(3) The standardized data is taken as input and entered into the $L_{1/2}$-SCN classifier for learning and training to obtain classification.

Algorithm 2

```
Set input-output data pair $(X, Y)$;
Initialize the parameters of LSTM, including learning rate, optimizer, activation function, weights and biases $W_i, W_f, W_c, W_o, b_i, b_f, b_c, b_o$; Optimization: SGDM with learning rate $\eta = 0.2$, momentum $\gamma = 0.5$
Calculate the outputs of LSTM according to Formula (1)-(6), and constantly update the weights by BP algorithm to get the output $h$;
Normalize $h$ to $h'$;
Input $h'$ into $L_{1/2}$-SCN classifier and perform calculation according to Algorithm 1;    ▷ Sigmoid activation function for all hidden nodes
Return the outputs of $L_{1/2}$-SCN: $y(L) = (y_1(L), y_2(L), ..., y_m(L))$.
```

The algorithm proposed above constitutes a two-stage hybrid approach. As an end-to-end learning framework, during the classification phase, the parameter selection for the $L_{1/2}$-SCN model is guided by a rigorous inequality supervision mechanism, and its convergence properties have been analyzed. Consequently, in comparison to other deep learning models, the proposed algorithm exhibits mathematical interpretability with respect to its convergence behavior. This is also the reason for this paper emphasizing the proposed model's some interpretability. However, we admit that the selection of model parameters is still random, and it is not a deterministic mathematical model that can be analyzed in terms of its underlying mechanism.

Numerical experiments

This section employs $L_{1/2}$-SCN on benchmark datasets to demonstrate its effectiveness in sparsity and generalization. The fusion algorithm is then used to determine the fault type based on the Case Western Reserve University dataset. Meanwhile, we also designed a comparative experiment using $L_{1/2}$-SCN without a feature extraction process to illustrate the effectiveness of feature extraction.

Experiments based on the benchmark datasets

The subsequent experiments rely on the Iris, Wine, Mnist, Prostate, and Dee datasets from UCI Machine Learning Repository. The first three datasets are used for classification, while the remaining datasets are used for regression. Table 1 summarizes the attributes of these datasets. In [27] and [28], the authors introduced SCN with L_2 and L_1 regularization terms, respectively, denoted as RSCN (Regularized SCN) and PSCN (Parsimonious SCN). The generalization performance and sparsity of $L_{1/2}$-SCN will be compared with RSCN, PSCN, and SCN. However, RSCN and SCN do not possess sparse capabilities, so $L_{1/2}$-SCN will primarily be compared with PSCN regarding sparsity. Table 2 reports the parameters of all models, where C represents the regularization parameter of RSCN.

Table 1. Attributes of dataset.

Dataset	Samples size	Training samples	Test samples	Input Features	Output Features	Attribute
Iris	120	90	30	4	3	Classification
Wine	178	148	30	13	3	Classification
Mnist	70000	60000	10000	784	10	Classification
Prostate	97	67	30	8	1	Regression
Dee	365	300	65	6	1	Regression

https://doi.org/10.1371/journal.pone.0339859.t001

Table 2. Parameters setting.

Dataset	Hidden nodes	λ	*C*	Iterations of ADMM
Iris	40	0.005	2^5	1000
Wine	20	0.005	2^{10}	1000
Mnist	200	0.0005	2^5	100
Prostate	200	0.0005	2^{10}	1000
Dee	70	0.005	2^{10}	2000

https://doi.org/10.1371/journal.pone.0339859.t002

Let N_R represent the samples that is correctly classified, N_T represent the total number of samples, the classification accuracy is defined as follows.

$$ACC = \frac{N_R}{N_T} \tag{42}$$

Define the root mean square error(RMSE) as follows.

$$RMSE = \sqrt{\left(\frac{1}{N_T}\sum_{i=1}^{N_T}(t^i - y^i)^2\right)} \tag{43}$$

where t^i is the target output of the i-th sample, while y^i is the network output. Let β_{total} represent the number of weights between the hidden layer and output layer ($L_{1/2}$-SCN, PSCN, RSCN and SCN), and let *D* represent the number of zero weights among them, and define sparsity *Z* as follows:

$$Z = \frac{\beta_{zero}}{\beta_{total}} \tag{44}$$

The regularization coefficient λ was examined via grid search. When λ was set to the values presented in Table 2, the optimal sparsity-precision trade-off was achieved. The number of hidden nodes was incrementally increased from one to the values listed in Table 2. Beyond these node counts, model performance remained stable.

Table 3 evaluates the models on the first three datasets based on classification accuracy and the last two datasets using the RMSE criterion. Therefore, the data in the table is described by ’Accuracy or RMSE’.

Table 3 reports the results of each test set, and Table 4 presents the sparsity of each model. Table 3 highlights that $L_{1/2}$-SCN exhibits superior generalization performance on most datasets, and its sparsity remains superior even when the generalization performance is comparable. In both the Iris and Prostate benchmark experiments, the classification accuracy progressively improves with increasing numbers of hidden nodes, while the regression error exhibits a consistent decline. This phenomenon demonstrates the critical role of node quantity in model capacity (Figs 4 and 5). Figs 6 and 7 demonstrate the weights distribution of the four models. Table 4 presents the sparsity, indicating that the sparsity degree of $L_{1/2}$-SCN is higher than that of PSCN, verifying that $L_{1/2}$ regularization leads to better sparsity.

Table 3. Performance comparison on the test set.

Dataset	Accuracy or RMSE			
	$L_{1/2}$-SCN	PSCN	RSCN	SCN
Iris	**96.33%**	93.00%	93.50%	90.50%
Wine	**99.67%**	99.50%	97.75%	96.50%
Mnist	**90.10%**	89.99%	89.58%	89.68%
Prostate	**0.2269**	0.2409	0.2373	0.9471
Dee	0.0932	0.0892	0.0892	0.0890

https://doi.org/10.1371/journal.pone.0339859.t003

Table 4. Sparsity of each model (Ratio of zero weights between the hidden layer and the output layer).

	$L_{1/2}$-SCN	PSCN	RSCN	SCN
Iris	**76.66%**	46.67%	0	0
Wine	**21.66%**	8.33%	0	0
Mnist	**4.70%**	0.05%	0	0
Prostate	**92.5%**	36.50%	0	0
Dee	**91.42%**	71.43%	0	0

https://doi.org/10.1371/journal.pone.0339859.t004

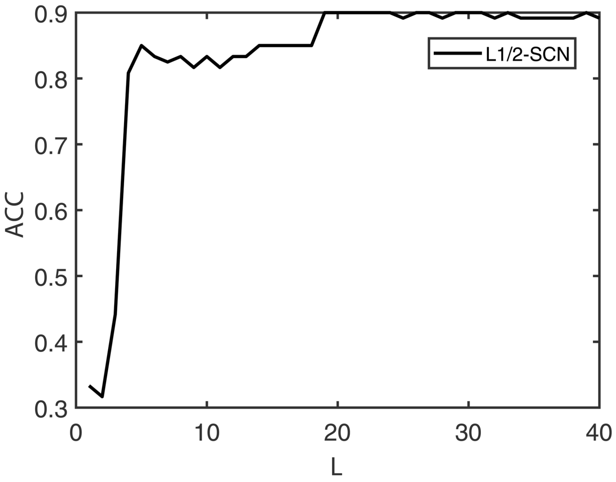


Fig 4. Convergence of L1/2-SCN: Training ACC achieve 98% with 40 nodes (Iris).

https://doi.org/10.1371/journal.pone.0339859.g004

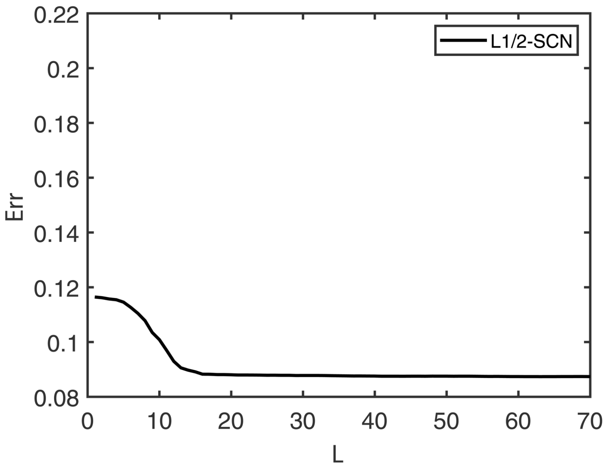


Fig 5. Convergence of L1/2-SCN: training loss drops below 0.09 with 70 nodes (Prostate).

https://doi.org/10.1371/journal.pone.0339859.g005

Notably, on the Mnist dataset, the model achieves increased accuracy as the number of hidden nodes rises to 200. Both $L_{1/2}$-SCN and PSCN demonstrate excellent performance on the Wine dataset, with an accuracy of less than 100% only one or two times out of 20 experiments. Notably, $L_{1/2}$-SCN excels in sparsity despite the similar classification capabilities of the two models.

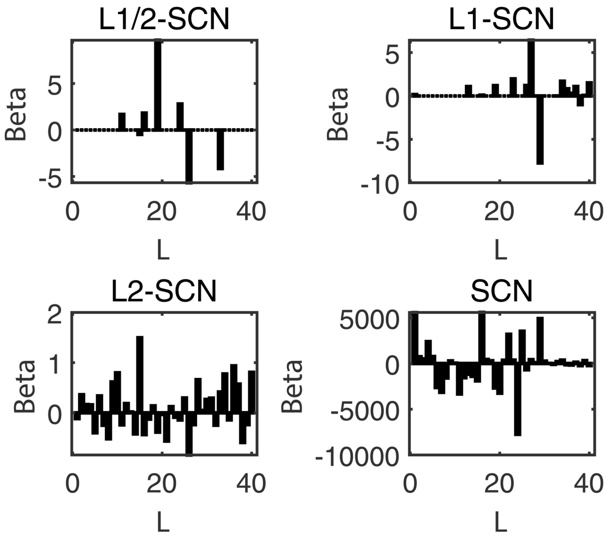


Fig 6. Sparsity pattern contrast: L1/2-SCN achieves 76.66% zero weights (Iris).

https://doi.org/10.1371/journal.pone.0339859.g006

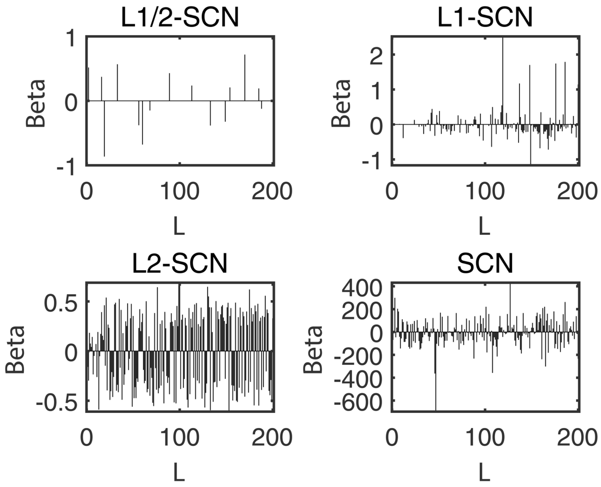


Fig 7. Sparsity pattern contrast: L1/2-SCN achieves 92.50% zero weights (prostate).

https://doi.org/10.1371/journal.pone.0339859.g007

Fault diagnosis experiment of rolling bearings

Experimental methodology. In this section, the performance of the proposed model is verified using the rolling bearing failure dataset of Western Reserve University in the United States. The rolling bearing fault experiment introduces varying-sized fault points into the three parts of the bearing. Precisely, accelerometers are placed on the bearing, the motor's driving terminus, and the fan-facing extremity to collect vibration data. Data from the motor housing drive end are also recorded at a sampling rate of 12,000 samples per second. Under a 12 kHz sampling rate, 12,000 samples correspond to a 1-second time duration, which fully encompasses the characteristic periodicity of typical bearing fault frequencies. This paper selects standard data and nine types of fault data spanning four cases of motors ranging from 0 to 3 horsepower (Cases 1 through 4). The fault points on the outer ring are located at 6 o'clock. The specific fault classifications are detailed in Table 5, and Fig 8 illustrates the vibration signals for nine different faults and normal operating states.

The specific experimental process is as follows. Step 1. Data Processing: The raw data is initially aligned to ensure consistency in length. For each category, the first 120,000 data points are selected. The signal data of length 120,000 is then divided into a matrix of 1200x100(The segment length of 100 points was determined through time-frequency analysis of bearing vibration characteristics), interpreted as 1200 samples. The supervised learning data for LSTM is constructed by considering the following 20 data points as their corresponding outputs(Through random forest MDI evaluation, the top 20 features are identified as critical discriminators, collectively accounting for 93.5% (95% CI: ±2.1%) of the importance weight), every 100 data points. Step 2. Feature extraction: LSTM extracts 1200*20 features for each category. Step 3. Dataset Splitting: The 1200 samples from each category in Step 2 are utilized for the second-stage experiment. One thousand samples are randomly selected to form the training set, while the remaining 200 constitute the test set.

Table 5. Classification of faults in motor housing driver-end data (case 4).

Status	Description
1	Manufacture a fault of 0.007 inches on the ball
2	Manufacture a fault of 0.014 inches on the ball
3	Manufacture a fault of 0.021 inches on the ball
4	Manufacture a fault of 0.007 inches in the inner ring
5	Manufacture a fault of 0.014 inches in the inner ring
6	Manufacture a fault of 0.021 inches in the inner ring
7	Manufacture a fault of 0.007 inches of the bearing outer ring at 6 o'clock
8	Manufacture a fault of 0.014 inches of the bearing outer ring at 6 o'clock
9	Manufacture a fault of 0.021 inches of the bearing outer ring at 6 o'clock
10	Normal state

https://doi.org/10.1371/journal.pone.0339859.t005

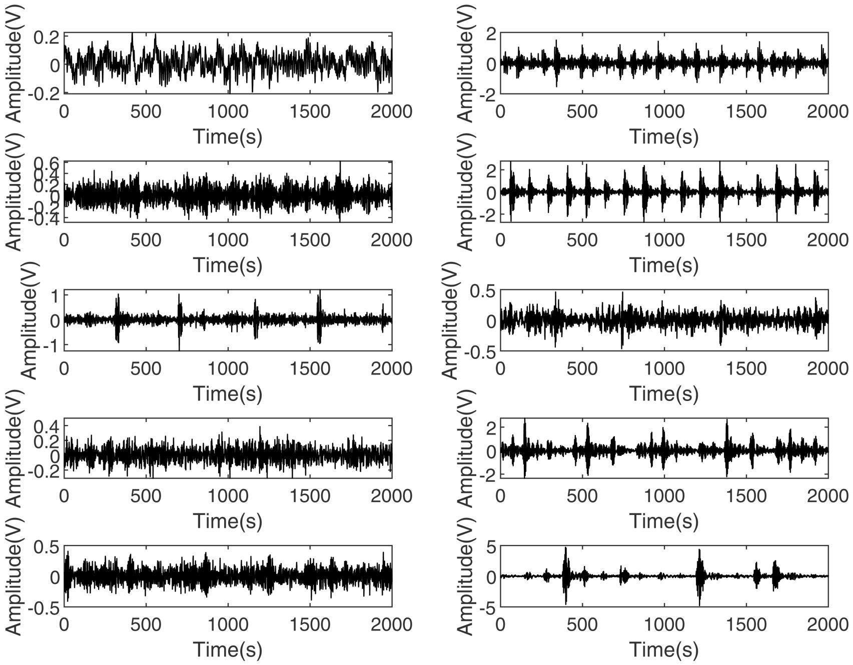


Fig 8. Fault data distribution diagram of the drive end of the motor housing.

https://doi.org/10.1371/journal.pone.0339859.g008

Consequently, the total number in the training and test sets for the second-stage experiment is 10,000 and 2,000, respectively. Step 4. The feature data obtained in Step 3 is normalized and then input into the $L_{1/2}$-SCN for classification, where the output represents the fault category. Fig 9 outlines the processing flow.

The fault identification ability of the proposed method is compared against Attention-TCN, Attention-BiLSTM, TCN-BiLSTM, Attention-TCN-BiLSTM, GRU, Resnet and TCN-Transformer models.

Evaluation indexes and results. The evaluation indicators are Test ACC, Precision, Recall, F1, AUC, ROC curve, PR curve. Considering binary classification, for example, TP signifies the number of True Positives, FP denotes the quantity of False Positives, and FN represents the number of False Negatives. These metrics are defined based on the TP, FP, TN, and FN.

$$Precision = \frac{TP}{TP + FP} \tag{45}$$

$$Recall = \frac{TP}{TP + FN} \tag{46}$$

$$F1 = \frac{2Precision * Recall}{Precision + Recall} \tag{47}$$

$$AUC = \frac{TP + TN}{TP + TN + FP + FN} \tag{48}$$

To ensure a comprehensive evaluation, compare LSTM-L1/2-SCN against a diverse set of benchmarks, which are selected to represent different architectural paradigms in time-series modeling and fault diagnosis.

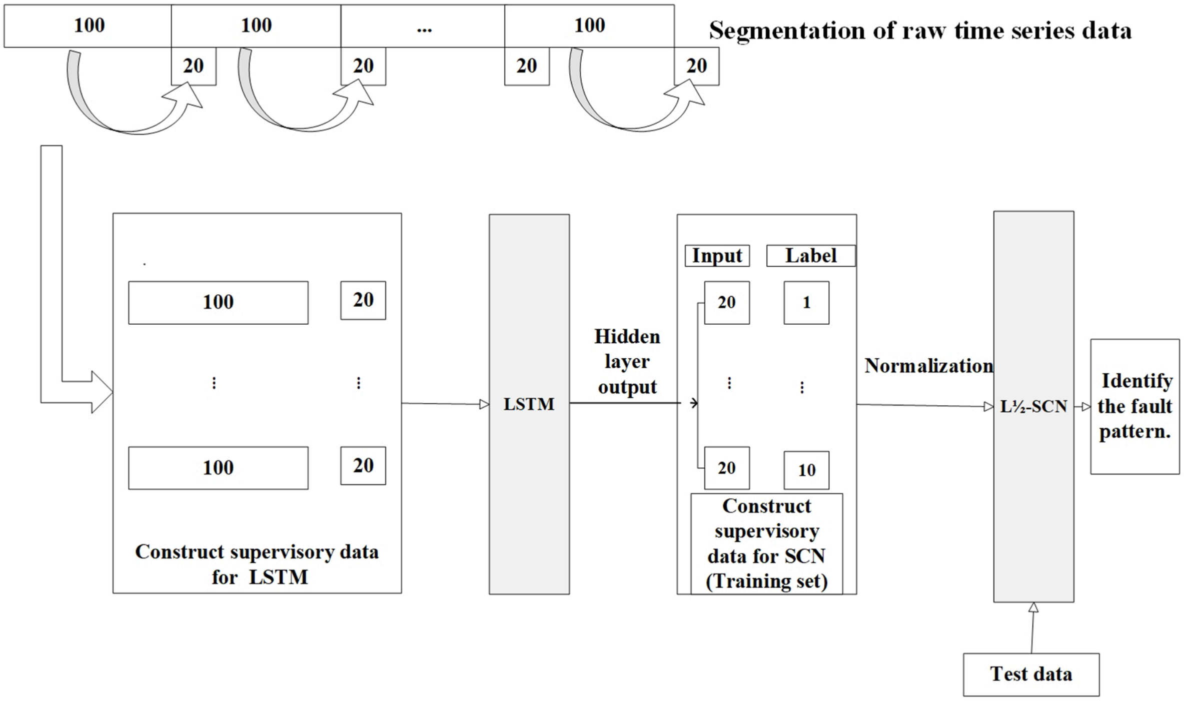

100
100
...
100
Segmentation of raw time series data
20
20
20
20
100
20
⋮
⋮
100
20
Construct supervisory data for LSTM
LSTM
Hidden layer output
Input
Label
20
1
⋮
⋮
20
10
Construct supervisory data for SCN (Training set)
Normalization
L½-SCN
Identify the fault pattern.
Test data

Fig 9. The processing flow of rolling bearing fault dataset.

https://doi.org/10.1371/journal.pone.0339859.g009

(1)Hybrid Models: Attention-TCN, Attention-BiLSTM, TCN-BiLSTM, and Attention-TCN-BiLSTM , which capture complex temporal dynamics by the trend of combining convolutional, recurrent and attention models.

(2)Sequential Model: GRU is selected as a simple basic baseline as recurrent neural networks out of these neural network types are known to efficiently work with sequential data.

(3)Deep Residual Architecture: ResNet, a model constructed for computer vision, is used to benchmark against a generic architecture that can learn complex hierarchical representations.

(4)Advanced Transformer-based Architecture: The TCN-Transformer model is selected for a contrastive impact with the long-range properties of a TCN and the global context learning capability of the Transformer, and is one of advanced architectures.

This selection guarantees that the proposed model is evaluated across a wide spectrum of technical routes, thereby providing a holistic demonstration of its performance.

Following the LSTM feature extraction, Tables 6 to 9 compare the performance between LSTM-$L_{1/2}$-SCN and Attention-TCN, Attention-BiLSTM, TCN-BiLSTM, Attention-TCN-BiLSTM, GRU, Resnet and TCN-Transformer. In the experiments, the procedure was executed 50 times.

The above results show that Attention - TCN - BiLSTM is a suboptimal model. To verify the significance of the proposed model in terms of performance comparison, a paired t-test was conducted between the proposed model and Attention - TCN - BiLSTM. Table 10 presents paired t-test results of $L_{1/2}$-SCN and Attention-TCN-BiLSTM.

To demonstrate the training process and sparsity effect of the proposed model, Figs 10 to 13 present the training convergence curves of the model under four cases, while Figs 14 to 17 show the weight distribution on the output side of $L_{1/2}$-SCN. Taking Case 1 as a representative instance, Figs 18 and 19 present the statistical indicators of the proposed model for each type of fault identification and their overall distribution, while Fig 20 presents the confusion matrix based on the test set. Figs 21 and 22 depict the ROC and PR curves, respectively.

Table 6. Comparison of experimental results (Case 1).

Model	Evaluation indexes (%, Mean ± 95% CI)				
	Test ACC	Precision	Recall	Macro F1	Macro AUC
LSTM-$L_{1/2}$-SCN	**0.9728±0.0096**	**0.9729±0.0096**	**0.9728±0.0096**	**0.9725±0.0100**	**0.9989±0.0003**
TCN-BiLSTM	0.9705±0.0008	0.9708±0.0008	0.9705±0.0008	0.9704±0.0008	0.9265±0.0022
Attention-TCN-BILSTM	0.9706±0.0009	0.9708±0.0008	0.9706±0.0009	0.9708±0.0008	0.9348±0.0017
Attention-TCN	0.9546±0.0012	0.9547±0.0012	0.9546±0.0012	0.9544±0.0013	0.9558±0.0013
Attention-BILSTM	0.9341±0.0012	0.9343±0.0012	0.9341±0.0012	0.9336±0.0012	0.9100±0.0008
GRU	0.8673±0.0014	0.8663±0.0014	0.8673±0.0014	0.8641±0.0016	0.9852±0.0002
Resnet	0.9509±0.0127	0.9552±0.0127	0.9509±0.0127	0.9507±0.0131	0.9964±0.0032
TCN-Transformer	0.9642±0.0017	0.9649±0.0014	0.9642±0.0017	0.9640±0.0017	0.8805±0.0047

https://doi.org/10.1371/journal.pone.0339859.t006

Table 7. Comparison of experimental results (Case 2).

Model	Evaluation indexes (%, Mean ± 95% CI)				
	Test ACC	Precision	Recall	Macro F1	Macro AUC
LSTM-$L_{1/2}$-SCN	**0.9751±0.0086**	**0.9750±0.0087**	**0.9751±0.0086**	**0.9748±0.0089**	**0.9991±0.0003**
TCN-BILSTM	0.9706±0.0009	0.9710±0.0008	0.9706±0.0008	0.9706±0.0009	0.9185±0.0026
Attention-TCN-BILSTM	0.9706±0.0008	0.9710±0.0009	0.9706±0.0008	0.9705±0.0009	0.9197±0.0022
Attention-TCN	0.9537±0.0013	0.9538±0.0013	0.9537±0.0013	0.9534±0.0013	0.9554±0.0013
Attention-BILSTM	0.9383±0.0010	0.9385±0.0011	0.9383±0.0010	0.9379±0.0010	0.9075±0.0010
GRU	0.8687±0.0013	0.8671±0.0013	0.8687±0.0013	0.8655±0.0014	0.9852±0.0002
Resnet	0.9559±0.0109	0.9572±0.0105	0.9559±0.0109	0.9557±0.0108	0.9976±0.0010
TCN-Transformer	0.9662±0.0014	0.9667±0.0014	0.9662±0.0014	0.9660±0.0015	0.8826±0.0043

https://doi.org/10.1371/journal.pone.0339859.t007

Table 8. Comparison of experimental results (Case 3).

Model	Evaluation indexes (%, Mean ± 95% CI)				
	Test ACC	Precision	Recall	Macro F1	Macro AUC
LSTM-$L_{1/2}$-SCN	**0.9710±0.0076**	**0.9712±0.0077**	**0.9710±0.0076**	**0.9708±0.0090**	**0.9978±0.0004**
TCN-BILSTM	0.9617±0.0011	0.9629±0.0008	0.9617±0.0011	0.9617±0.0010	0.9500±0.0017
Attention-TCN-BILSTM	0.9638±0.0007	0.9643±0.0007	0.9638±0.0007	0.9637±0.0007	0.9489±0.0013
Attention-TCN	0.9496±0.0010	0.9500±0.0009	0.9496±0.0010	0.9495±0.0009	0.9684±0.0011
Attention-BILSTM	0.9459±0.0010	0.9463±0.0009	0.9459±0.0010	0.9457±0.0010	0.908±0.0017
GRU	0.9248±0.0013	0.9256±0.0012	0.9248±0.0013	0.9244±0.0014	0.9939±0.0001
Resnet	0.9553±0.0210	0.9565±0.0182	0.9553±0.0210	0.9547±0.0222	0.9976±0.0014
TCN-Transformer	0.9616±0.0023	0.9627±0.0021	0.9616±0.0023	0.9615±0.0076	0.9051±0.0041

https://doi.org/10.1371/journal.pone.0339859.t008

Table 9. Comparison of experimental results (Case 4).

Model	Evaluation indexes (%, Mean ± 95% CI)				
	Test ACC	Precision	Recall	Macro F1	Macro AUC
LSTM-$L_{1/2}$-SCN	**0.9680±0.0091**	**0.9685±0.0092**	**0.9680±0.0091**	**0.9679±0.0100**	0.9899±0.0003
TCN-BILSTM	0.9624±0.0009	0.9628±0.0007	0.9625±0.0009	0.9624±0.0009	0.9195±0.0019
Attention-TCN-BILSTM	0.9626±0.0008	0.9630±0.0008	0.9626±0.0008	0.9625±0.0008	0.9295±0.0017
Attention-TCN	0.9517±0.0009	0.9519±0.0009	0.9517±0.0009	0.9516±0.0009	0.9553±0.0016
Attention-BILSTM	0.9285±0.0012	0.9287±0.0010	0.9285±0.0012	0.9280±0.0011	0.9004±0.0016
GRU	0.8615±0.0015	0.8636±0.0016	0.8615±0.0015	0.8599±0.0016	0.9842±0.0002
Resnet	0.9375±0.0323	0.9414±0.0280	0.9375±0.0323	0.9377±0.0317	**0.9966±0.0020**
TCN-Transformer	0.9609±0.0013	0.9616±0.0012	0.9609±0.0013	0.9607±0.0013	0.8806±0.0048

https://doi.org/10.1371/journal.pone.0339859.t009

Table 10. Paired t-test results of $L_{1/2}$-SCN vs. Attention TCN-BiLstm (Case 1).

Statistical metric	t-value	p-value
Test ACC	3.8833	0.0003
Precision	3.4725	0.0011
Recall	3.8833	0.0003
Macro F1	3.9434	0.0003
Macro AUC	79.0984	0.0000

https://doi.org/10.1371/journal.pone.0339859.t010

To showcase the efficacy of LSTM in feature abstraction and extraction, LSTM-$L_{1/2}$-SCN is compared with $L_{1/2}$-SCN without data feature extraction, with the corresponding fault identification results reported in Table 11. The test accuracy is less than 60%, which infer the important role played by the LSTM model in feature extraction in the first stage.

In order to analyze the influence of the value of theegular parameter λ on the sparsity and performance of the model, Table 12 lists the results of LSTM-$L_{1/2}$-SCN when the regular parameters are set to 0.005 and 0.01, respectively. The results infer that the regularization coefficient λ significantly impacts the sparsity of $L_{1/2}$-SCN. The larger λ, the stronger the sparsity. Therefore, choosing the appropriate coefficient requires a parameter-tuning process to balance the two aspects.

Table 13 presents the sparsity of LSTM-$L_{1/2}$-SCN across the four working conditions, and Table 14 presents a computational cost comparison between the LSTM-$L_{1/2}$-SCN model and the Attention-TCN-BiLSTM model.

To further verify the generalization performance of the model, Table 15 validates the performance of the proposed model based on noisy datasets (with noise added).

To highlight the overall merits of the proposed method, Table 16 compares its performance with other models in terms of sparsity and classification accuracy. The values reported represent the average experimental results across four operating conditions derived from the CWRU dataset, whereas the accuracy of competing models is averaged based on their suboptimal experimental outcomes.

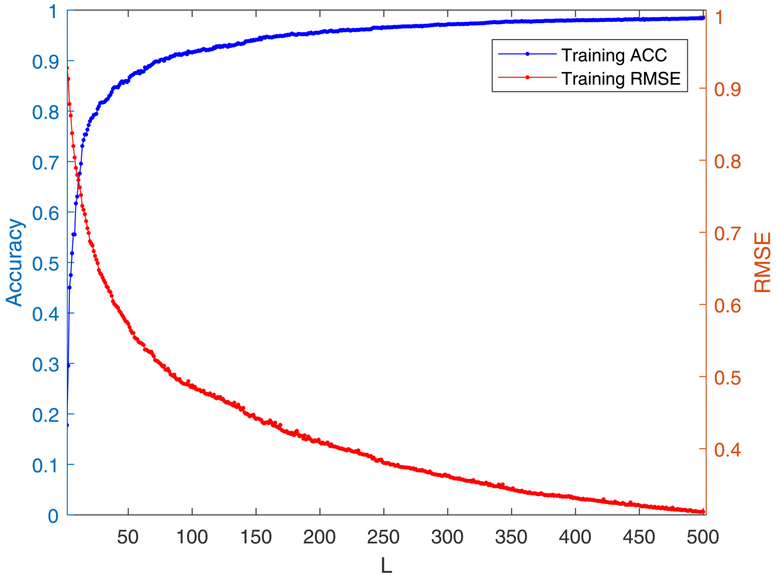


Fig 10. **Convergence of L1/2-SCN: training accuracy exceeds 98% with 500 hidden nodes in Case 1.**

https://doi.org/10.1371/journal.pone.0339859.g010

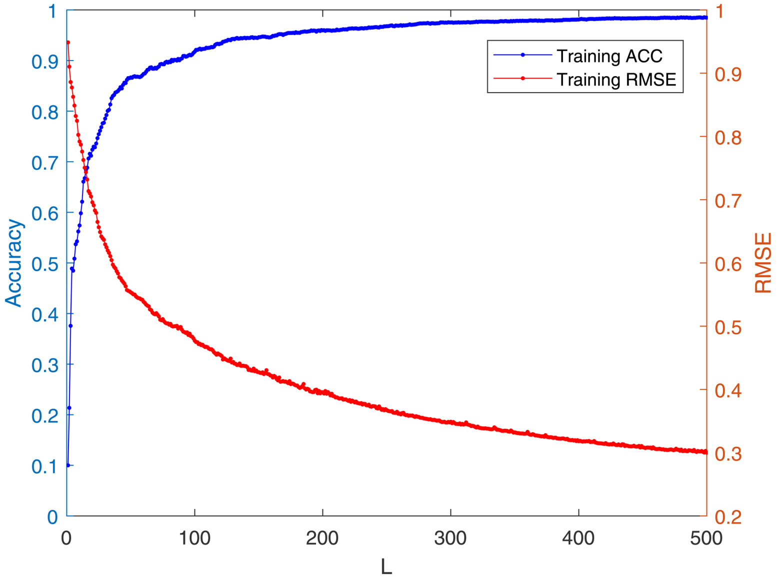


Fig 11. **Training accuracy exceeds 98% with 500 hidden nodes in Case 2.**

https://doi.org/10.1371/journal.pone.0339859.g011

Results analysis and discussion

Benchmark experiments. The results on five benchmark datasets (Tables 3 and Table 4, Figs 4 to 7) demonstrate that $L_{1/2}$-SCN exhibits superior sparsity and generalization capabilities. Regarding sparsity performance, when compared to PSCN, $L_{1/2}$-SCN has a maximum increase of 56%. This is primarily due to integrating $L_{1/2}$ regularization into SCN, which offers better sparsity than the L_1 regularization. It also results in a significant number of zero weights, effectively preventing overfitting and enhancing the generalization ability.

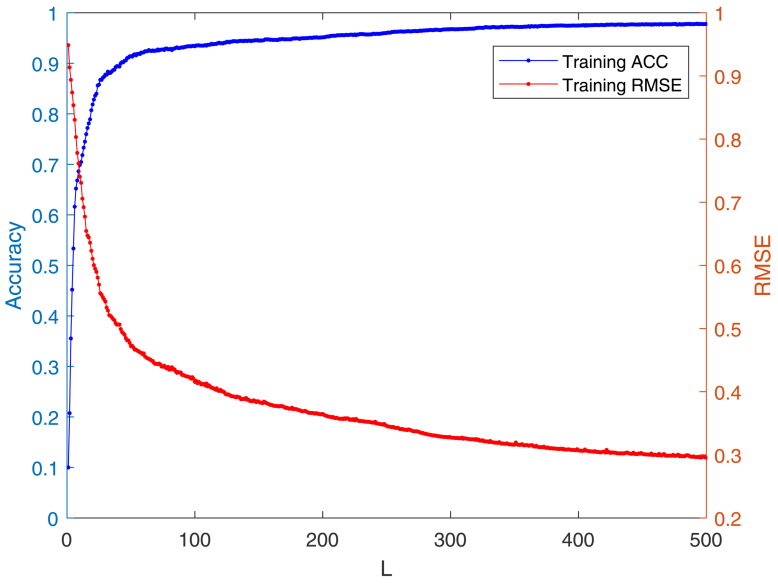


Fig 12. Training accuracy exceeds 98% with 500 hidden nodes in Case 3.

https://doi.org/10.1371/journal.pone.0339859.g012

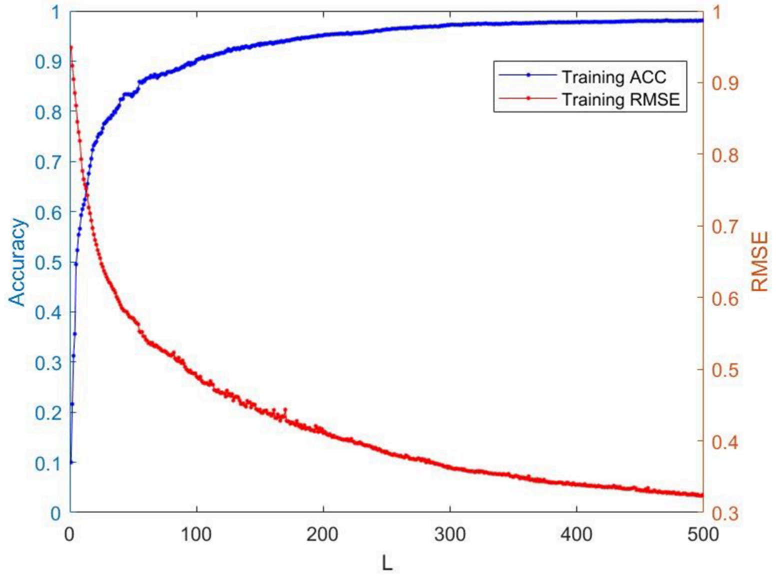


Fig 13. Training accuracy exceeds 98% with 500 hidden nodes in Case 4.

https://doi.org/10.1371/journal.pone.0339859.g013

Fault diagnosis experiments. Statistics metrics analysis: Tables 6 to 9 show that LSTM-$L_{1/2}$-SCN performs exceptionally well on the mean values of all five indicators. Take condition 1 as an example. The Test ACC of LSTM-$L_{1/2}$-SCNis 97.28%, which is 0.22 percentage points higher than that of the suboptimal model Attention-TCN-BiLSTM. The Precision is 0.9729, which is 0.21 percentage points higher than that of the suboptimal model. The Recall is 0.9728, which is 0.21 percentage points higher than that of the suboptimal model. F1 is 0.9725, which is 0.17 percentage points higher than that of the suboptimal model. The AUC is 0.9989, which is 6.41 percentage points higher than that of the suboptimal

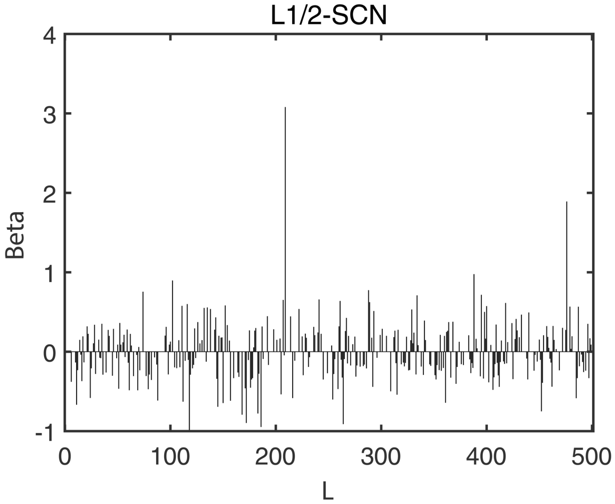


Fig 14. Sparsity pattern contrast: L1/2-SCN achieves 24.24% zero weights (Case 1).

https://doi.org/10.1371/journal.pone.0339859.g014

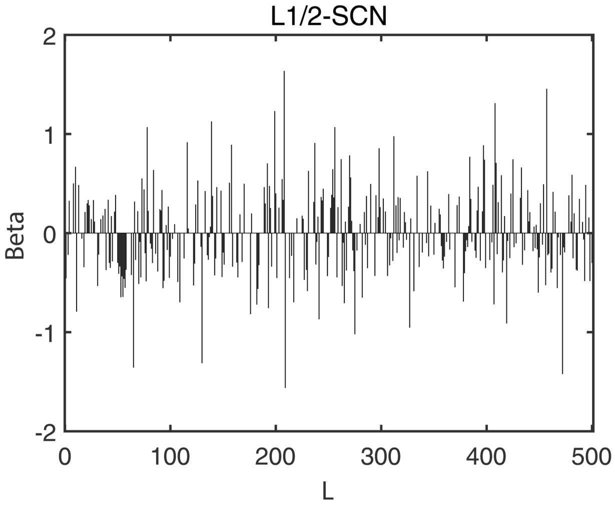


Fig 15. Sparsity pattern contrast: L1/2-SCN achieves 23.88% zero weights (Case 2).

https://doi.org/10.1371/journal.pone.0339859.g015

model. These indicators illustrate the superiority of the model in terms of accuracy. However, the drawback of LSTM-$L_{1/2}$-SCN is that the variance of the experimental results is relatively large, which indicates that the randomness of the model parameter values is still relatively high and requires subsequent improvement. Figs 10 to 13 show the training convergence curves of the proposed model for one experiment conducted under each of the four working conditions. It can be seen that when the number of hidden layer nodes increases to 500, the accuracy rate of the model on the training set can all exceed 98%, indicating the good performance of the model. Figs 14 to 17 present the weight distribution of $L_{1/2}$-SCN in the fusion model, the percentage of zero weight is above 23%, verifying the sparse effect.

Confusion matrix analysis: In order to observe the intuitive recognition of various types of faults by the model, Fig 20 presents the confusion matrix of the test set in a certain experiment. It can be seen that the classification effect of the model for categories 8 and 10 is not good. These two categories are ’a 0.014-inch fault on the bearing outer ring at 6

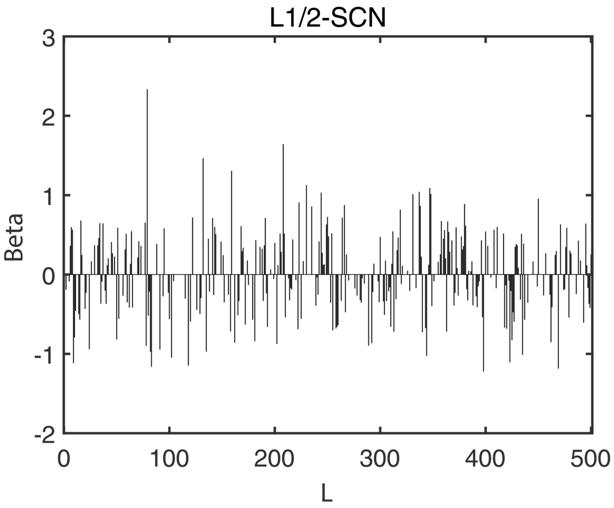


Fig 16. Sparsity pattern contrast: L1/2-SCN achieves 29.39% zero weights (Case 3).

https://doi.org/10.1371/journal.pone.0339859.g016

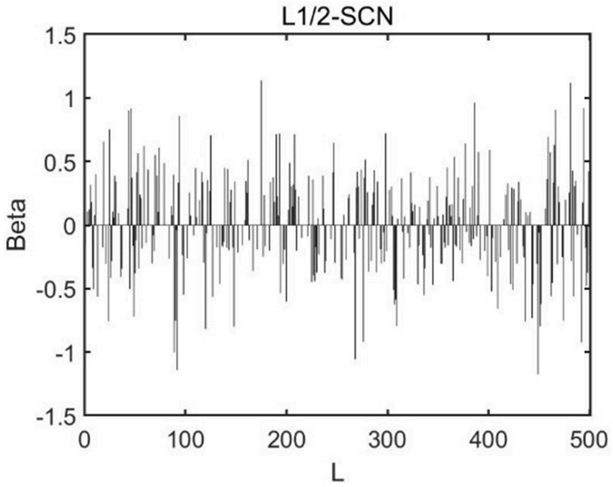


Fig 17. Sparsity pattern contrast: L1/2-SCN achieves 24.72% zero weights (Case 4).

https://doi.org/10.1371/journal.pone.0339859.g017

o'clock' and 'Normal state'. This is related to the data distribution and quality to some extent. As can be seen from Fig 8, the vibration periodicity of the 8th type of data is poor and the data variance is large, while the 10th type of data is affected by some outliers (noise). This will affect the learning effect of the model and thereby the classification effect.

The paired t-test analysis (Table 10): The results of the paired t-test reveal statistically significant disparities between the proposed model and the a-cnn-bilstm model across five pivotal performance metrics: test accuracy (test acc), precision, recall, macro F1 score, and macro AUC. Specifically, for all five metrics, the t-statistics exhibit relatively high values, with corresponding p-values substantially below the conventional significance threshold of 0.05. This finding underscores that the observed performance differences between the two models are unlikely to be attributable to random variation and are instead statistically robust. Notably, in the macro AUC metric, the t-statistic reaches an exceptionally high value of 79.0984, accompanied by a p-value approaching zero. This compelling evidence further substantiates that the proposed model outperforms the a-cnn-bilstm model markedly in discriminating between positive and negative samples.

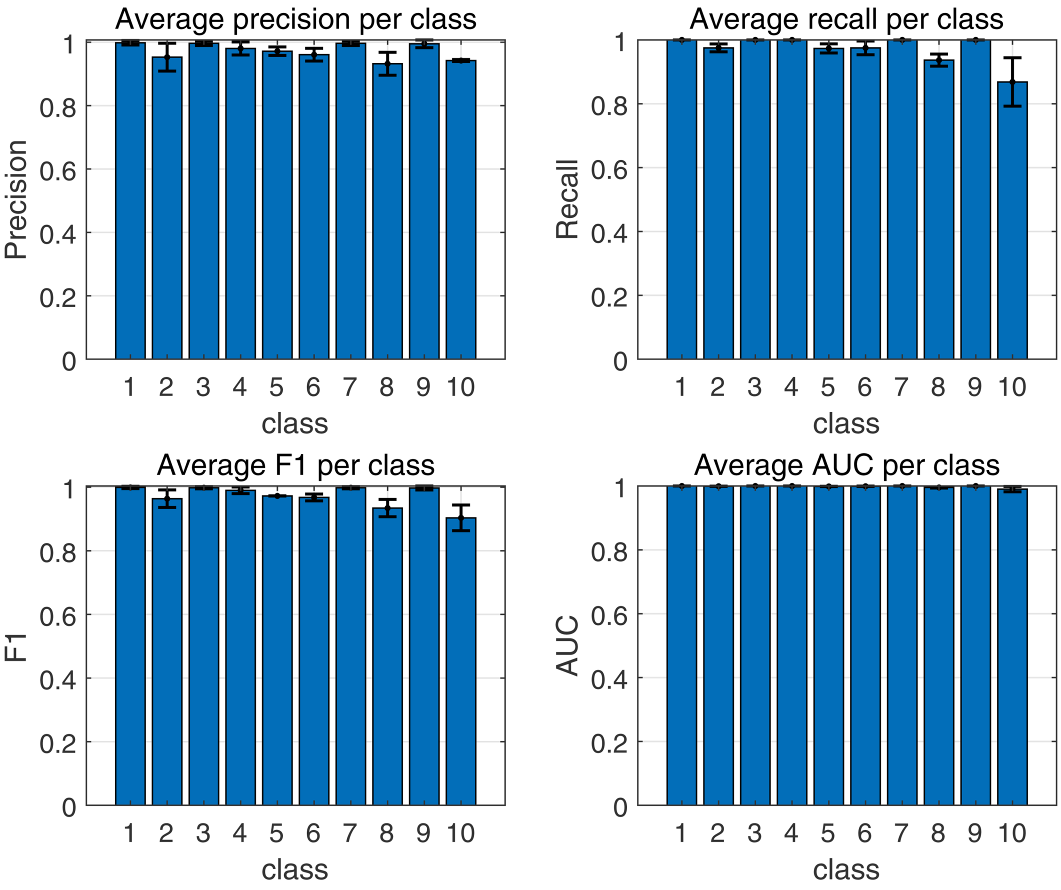


Fig 18. Average metric per class (Case 1).

https://doi.org/10.1371/journal.pone.0339859.g018

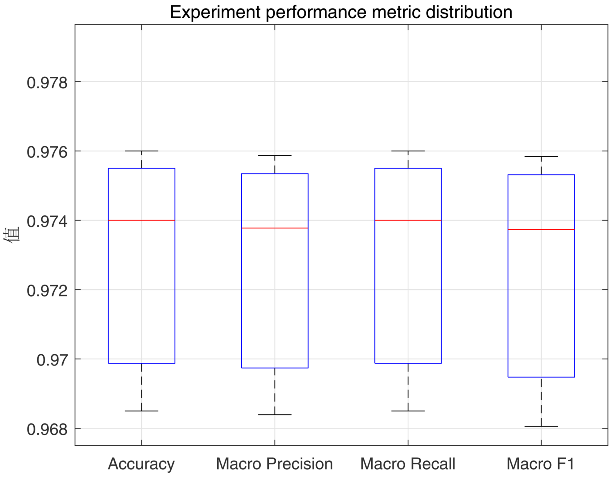
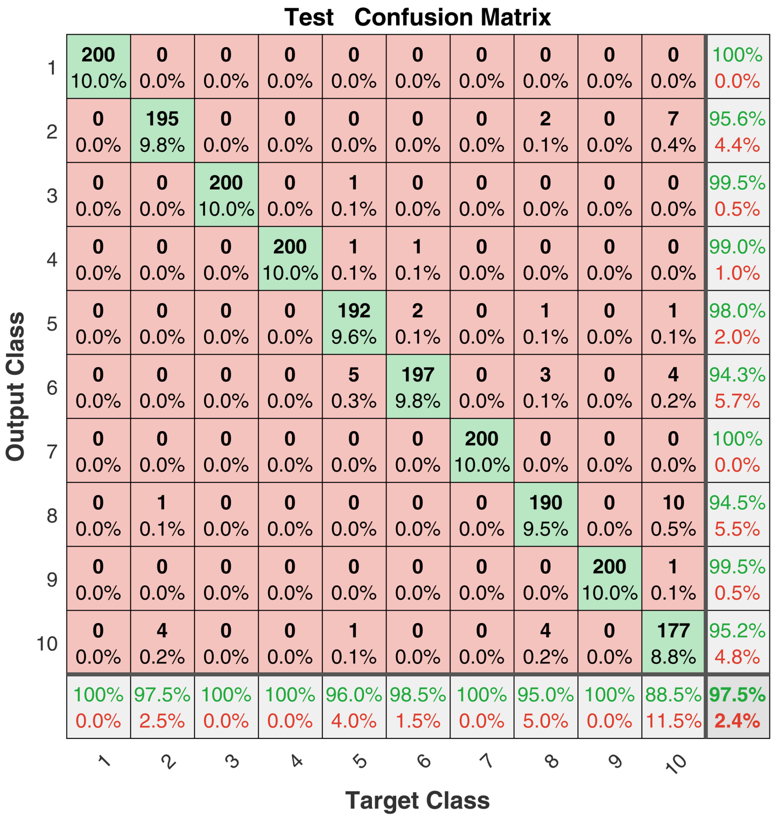


Fig 19. Experiment performance metric distribution (Case 1).

https://doi.org/10.1371/journal.pone.0339859.g019



Test Confusion Matrix

Output Class \ Target Class	1	2	3	4	5	6	7	8	9	10	
1	200 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	195 9.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.1%	0 0.0%	7 0.4%	95.6% 4.4%
3	0 0.0%	0 0.0%	200 10.0%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.5% 0.5%
4	0 0.0%	0 0.0%	0 0.0%	200 10.0%	1 0.1%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.0% 1.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	192 9.6%	2 0.1%	0 0.0%	1 0.1%	0 0.0%	1 0.1%	98.0% 2.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 0.3%	197 9.8%	0 0.0%	3 0.1%	0 0.0%	4 0.2%	94.3% 5.7%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	200 10.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
8	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	190 9.5%	0 0.0%	10 0.5%	94.5% 5.5%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	200 10.0%	1 0.1%	99.5% 0.5%
10	0 0.0%	4 0.2%	0 0.0%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	4 0.2%	0 0.0%	177 8.8%	95.2% 4.8%
	100% 0.0%	97.5% 2.5%	100% 0.0%	100% 0.0%	96.0% 4.0%	98.5% 1.5%	100% 0.0%	95.0% 5.0%	100% 0.0%	88.5% 11.5%	**97.5%** **2.4%**

Fig 20. The confusion matrix on the test set for a specific experiment (Case 1).

https://doi.org/10.1371/journal.pone.0339859.g020

ROC and PR curve analysis (Figs 21 and 22): The ROC curves closely approach the top-left corner, with a minimum AUC value of 0.98 and an average AUC exceeding 0.99 across all samples, indicating high classification accuracy across all thresholds. As evidenced by a well-behaved ROC curve indicating stable model performance. Similarly, the PR curves, boast a minimum of 0.91 and an average of 0.987, highlighting the model's balance between accuracy and recall rates, underscoring its better performance.

Analysis of the Feature Extraction Function of LSTM (Table 11): The first stage uses only the basic LSTM model to maintain model simplicity, avoiding more complex alternatives like BiLSTM. A comparative experiment underscores LSTM's role in feature extraction, contrasting the $L_{1/2}$-SCN model without LSTM. In Table 11, the average test ACC of $L_{1/2}$-SCN without LSTM under the same parameter settings is 48.06%. Therefore, the shallow model alone exhibits limitations in handling large-scale data, with suboptimal fault identification performance using solely $L_{1/2}$-SCN. Hence, fusing LSTM and $L_{1/2}$-SCN can better realize fault identification.

Sparsity and the regularization parameter λ analysis (Tables 12 and 13): Table 13 highlights that the sparsity of $L_{1/2}$-SCN surpasses 23% when the regularization coefficient λ is set to 0.005, and it can even exceed 33% when the coefficient increases to 0.01. This indicates that SCN attains better sparsity when enhanced with the $L_{1/2}$ regularization technique. Although a larger regularization coefficient generally leads to better sparsity, striking a balance is crucial, as excessively sparse models can result in reduced effective weights, ultimately compromising the model's accuracy. Therefore, we need to select an appropriate value for λ. Based on the results of the sensitivity experiment in this paper, λ is selected as 0.005. Figs 14 to 17 illustrate the weight distribution, revealing that the zero weights are almost uniformly generated during the gradual increase of the model's hidden units, which is determined by the principle of $L_{1/2}$ regularization and is

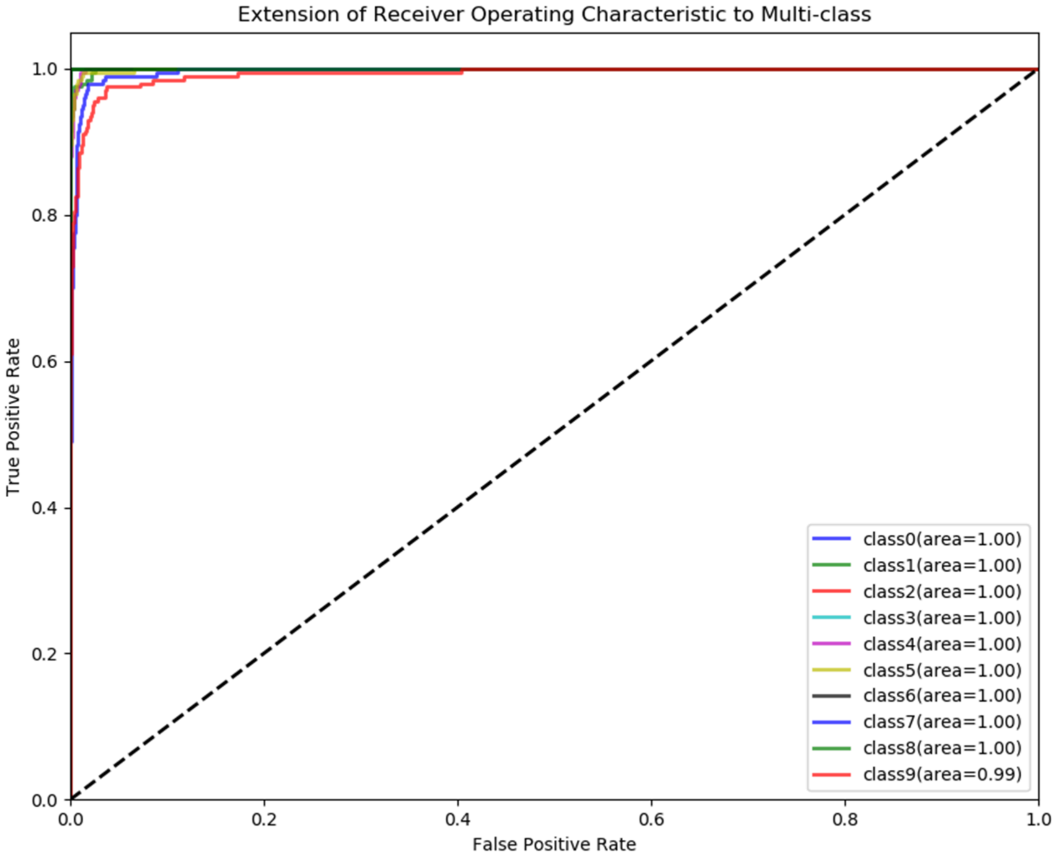


Fig 21. Multi-class discriminability: receiver operating characteristic (ROC) curves for LSTM-L1/2-SCN model with all AUC >0.99 (Case 1).

https://doi.org/10.1371/journal.pone.0339859.g021

also in line with our expectations. In summary, $L_{1/2}$-SCN is used for fault identification in the second stage, affording better sparsity. On the test set, this model achieves an accuracy of 97.20% while maintaining a sparsity level exceeding 23%. Compared to current deep learning models, such as TCN and LSTM, the proposed approach exhibits distinct advantages in terms of sparsity, proving the validity of the $L_{1/2}$-SCN fusion sparse algorithm.

Computational Cost (Table 14): Table 14 reveals that the LSTM-$L_{1/2}$-SCN model demonstrates a notable computational efficiency advantage, with training time reduced to 1/21.8 of the attention mechanism-based temporal model (9.25 minutes per training session), peak memory usage controlled at 2.5GB (58% lower than the 6GB of the comparative model), floating-point operations decreased by 53%, and parameter size only 10% of the comparative model; this efficiency stems from triple optimization—LSTM sequence modeling avoiding large convolution kernel calculations, $L_{1/2}$ regularization eliminating redundant connections via sparse constraints, and an incremental node growth mechanism dynamically adjusting network complexity—making it suitable for deployment in edge computing units of resource-constrained industrial equipment.

Generalization Ability Analysis (Table 15): To verify the robustness and generalization performance of the model, Gaussian noise with a zero - mean and a standard deviation of 0.05, as well as uniformly distributed perturbations with an amplitude range of [-0.05, 0.05], were added, increasing the diversity of the data set. Table 15 presents the various statistical indicators for the model's classification of the new data set. It can be seen that when the model processes the data after adding noise, in terms of Test ACC, Precision, Recall, and F1, the indicators have decreased on average by approximately 0.2 percentage points, and in terms of AUC, they have decreased by 9 percentage points. However, even when

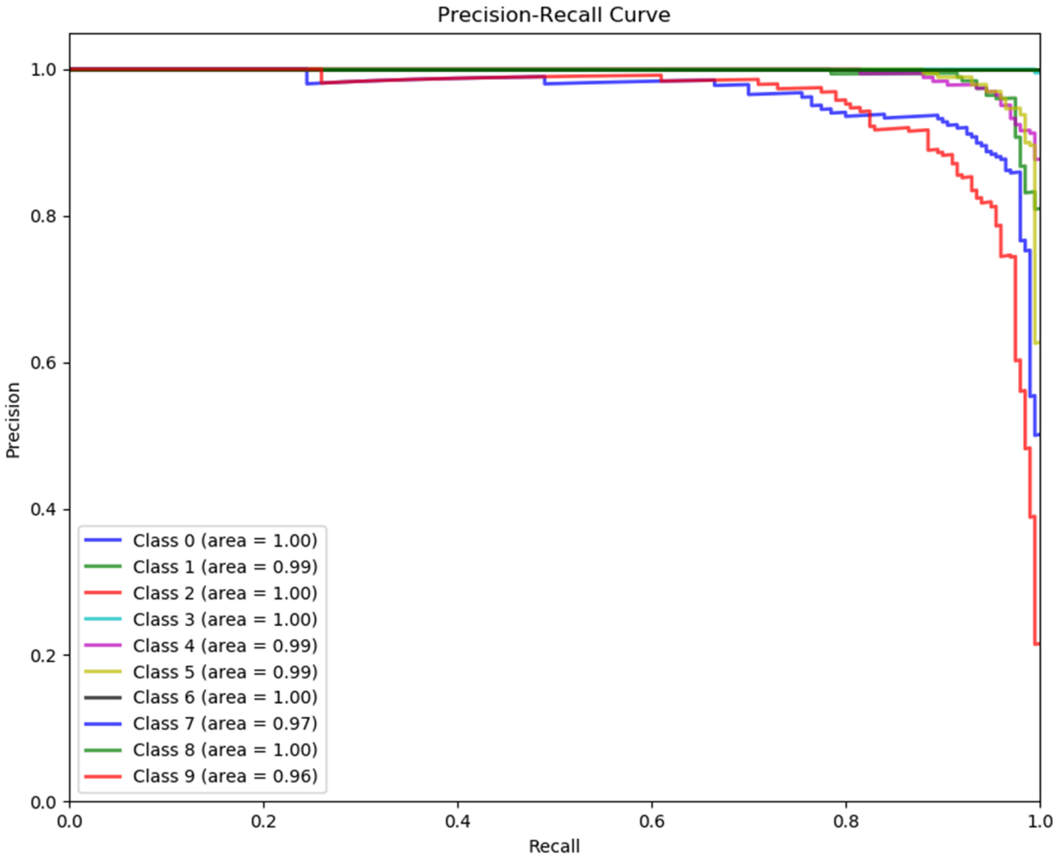


Fig 22. Precision-Recall Dominance: Class-wise curves exhibiting minimum AUPRC of 0.96 with five classes achieving perfection (Case 1).

https://doi.org/10.1371/journal.pone.0339859.g022

Table 11. Experimental results of L$_{1/2}$-SCN without LSTM.

Case	Training ACC	Test ACC	Sparsity
Case 1	39.62%	35.05%	9.70%
Case 2	63.30%	57.95%	9.68%
Case 3	61.38%	54.25%	10.30%
Case 4	51.44%	45.00%	9.58%

https://doi.org/10.1371/journal.pone.0339859.t011

Table 12. Results of fault identification of rolling bearings with different regularization coefficient.

Case	Training ACC, Test ACC	
	$\lambda = 0.005$	$\lambda = 0.01$
Case1	98.46%, 97.20%	98.51%, 97.00%
Case2	98.59%, 97.50%	98.73%, 98.15%
Case3	97.81%, 97.10%	98.43%, 97.30%
Case4	98.17%, 96.80%	98.06%, 96.55%

https://doi.org/10.1371/journal.pone.0339859.t012

exposed to noise, the model can still maintain a level of more than 90% in key performance indicators. This indicates that when facing a certain degree of data change, the core classification and prediction capabilities of the model have not been fundamentally damaged.

Table 13. Sparsity of L$_{1/2}$-SCN.

Case	λ = 0.00	λ = 0.01
Case1	24.24%	33.62%
Case2	23.88%	34.18%
Case3	29.39%	33.82%
Case4	24.72%	34.50%

https://doi.org/10.1371/journal.pone.0339859.t013

Table 14. Computational cost comparison.

Metrics	LSTM-L$_{1/2}$-SCN	Attention-Tcn-BiLSTM	Ratio
Training time	9.25min	198min	1:21.8
Resource usage	2.5G	6G	1:2.4
FLOPs	74.6G	160.16G	1:2.15
Params	15.9K	157K	1:10

https://doi.org/10.1371/journal.pone.0339859.t014

Table 15. Comparison of LSTM-L$_{1/2}$-SCN performance before and after adding noise to the dataset (Case 1).

Model	Evaluation indexes (%, Mean ± 95% CI)				
	Test ACC	Precision	Recall	Macro F1	Macro AUC
Before adding noise	**0.9728±0.0096**	**0.9728±0.0096**	**0.9727±0.0096**	**0.9725±0.0100**	**0.9989±0.0003**
After adding noise	0.9706±0.0008	0.9710±0.0009	0.9706±0.0008	0.9705±0.0009	0.9197±0.0022

https://doi.org/10.1371/journal.pone.0339859.t015

Table 16. Comparison and summary of LSTM-L$_{1/2}$-SCN and others.

	LSTM-L$_{1/2}$-SCN	Others
Sparsity	25.56%	N/A
Accuracy	97.17%	96.69%

https://doi.org/10.1371/journal.pone.0339859.t016

In conclusion, the enhanced performance of the proposed method stems primarily from its unique model structure, differing from conventional deep models. The seven deep learning models in the comparison perform end-to-end tasks, integrating two steps into one process. However, their feature mapping lacks a theoretical foundation. On the contrary, the proposed approach employs LSTM as a feature extractor, effectively condensing the original data while preserving historical time information. This compressed data is input into a shallow model, SCN, leveraging its universal approximation capability. We further refine SCN's structure with L$_{1/2}$ regularization, enhancing conciseness and minimizing redundancy. This two-stage learning model exhibits sparsity and achieves higher accuracy, leading to improved fault identification results.

Conclusion

This study presents an integrated LSTM and L$_{1/2}$-SCN architecture for rolling bearing fault diagnosis. By fusing temporal feature extraction with non-convex sparse regularization, the model achieves 25.56% weight sparsity (achieves an average improvement of 24.8% over PSCN) while reducing training duration by 95.3%. Convergence is guaranteed through a reconstructed supervision mechanism validated by mathematical formulas. Testing on the CWRU 10-class dataset yields 97.17% accuracy - surpassing comparable deep models by 0.2-10 percentage points. The implementation demonstrates industrial viability by enabling real-time diagnosis, which is suitable for edge deployment in rotating machinery monitoring systems.

Nevertheless, the model exhibits limitations under extreme variable operating conditions, particularly in multi-fault coupling scenarios. These constraints originate from the inherent non-stationarity of vibration signals and the current feature extraction mechanism's limited frequency band adaptability.

Future work will focus on exploring the comprehensive integration of multimodal information to further enhance the modeling and prediction capabilities in complex scenarios. Specifically, the idea of integrating multi-scale time series modules for prediction [33] and the relational interaction modeling method [34] can be applied. Meanwhile, this work will explore the architectural design of a modal fusion Vision Transformer (ViT), similar to [35], and the multimodal deep learning scheme outlined in [36], and study the fusion strategies for lightweight and adaptive models. This direction aims to build a more flexible multimodal fusion system to solve complex problems involving multi-source heterogeneous data. To ensure the practical deployment of such advanced systems, future work will also involve benchmarking the models on specific edge platforms and evaluating key metrics such as inference latency and power consumption.

Acknowledgments

The authors would like to express their gratitude to EditSprings (https://www.editsprings.cn) for the expert linguistic services provided.

Author contributions

Conceptualization: Jingjing Liu.

Formal analysis: Yanwei Ma.

Methodology: Yefeng Liu, Jingjing Liu.

Software: Yefeng Liu, Yanwei Ma, Shuai Wang.

Writing – original draft: Yefeng Liu, Jingjing Liu, Yanwei Ma.

Writing – review & editing: Jingjing Liu, Qichun Zhang.

References

1. Lei Y, Jia F, Lin J, Xing S, Ding SX. An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data. IEEE Trans Ind Electron. 2016;63(5):3137–47. https://doi.org/10.1109/tie.2016.2519325
2. Guan T, Liu S, Xu W, Li Z, Huang H, Wang Q. Rolling bearing fault diagnosis based on component screening vector local characteristic-scale decomposition. Shock and Vibration. 2022;2022:1–13. https://doi.org/10.1155/2022/9925681
3. Fang Z, Wu Q-E, Wang W, Wu S. Research on improved fault detection method of rolling bearing based on signal feature fusion technology. Applied Sciences. 2023;13(24):12987. https://doi.org/10.3390/app132412987
4. Zhang Y, Ji JC, Ren Z, et al. Digital twin-driven partial domain adaptation network for intelligent fault diagnosis of rolling bearing. Reliab.Eng.Syst.Saf. 2023; 234: 109186.
5. Zhang C, Qin F, Zhao W, Li J, Liu T. Research on rolling bearing fault diagnosis based on digital twin data and improved ConvNext. Sensors (Basel). 2023;23(11):5334. https://doi.org/10.3390/s23115334 PMID: 37300061
6. Chen Z, Li C, Sanchez R-V. Gearbox fault identification and classification with convolutional neural networks. Shock and Vibration. 2015;2015:1–10. https://doi.org/10.1155/2015/390134
7. Shao H, Jiang H, Zhang H, Duan W, Liang T, Wu S. Rolling bearing fault feature learning using improved convolutional deep belief network with compressed sensing. Mechanical Systems and Signal Processing. 2018;100:743–65. https://doi.org/10.1016/j.ymssp.2017.08.002
8. Wen L, Li X, Gao L, Zhang Y. A new convolutional neural network-based data-driven fault diagnosis method. IEEE Trans Ind Electron. 2018;65(7):5990–8. https://doi.org/10.1109/tie.2017.2774777
9. Wang G, Li Y, Wang Y, Wu Z, Lu M. Bidirectional shrinkage gated recurrent unit network with multiscale attention mechanism for multisensor fault diagnosis. IEEE Sensors J. 2023;23(20):25518–33. https://doi.org/10.1109/jsen.2023.3307729
10. Cui Q, Zhu L, Feng H, He S, Chen J. Intelligent fault quantitative identification via the improved Deep Deterministic Policy Gradient (DDPG) algorithm accompanied with imbalanced sample. IEEE Trans Instrum Meas. 2023;72:1–13. https://doi.org/10.1109/tim.2023.3250284
11. Chen X, Zhang B, Gao D. Bearing fault diagnosis base on multi-scale CNN and LSTM model. Intell. Manuf. 2021; 32: 971–87.
12. Sinitsin V, Ibryaeva O, Sakovskaya V, Eremeeva V. Intelligent bearing fault diagnosis method combining mixed input and hybrid CNN-MLP model. Mechanical Systems and Signal Processing. 2022;180:109454. https://doi.org/10.1016/j.ymssp.2022.109454
13. Liu Z, Wang H, Liu J, Qin Y, Peng D. Multitask learning based on lightweight 1DCNN for fault diagnosis of wheelset bearings. IEEE Trans Instrum Meas. 2021;70:1–11. https://doi.org/10.1109/tim.2020.3017900

14. Han T, Tian Z, Yin Z, Tan ACC. Bearing fault identification based on convolutional neural network by different input modes. J Braz Soc Mech Sci Eng. 2020;42(9). https://doi.org/10.1007/s40430-020-02561-6

15. Han S, Jeong J. An weighted CNN ensemble model with small amount of data for bearing fault diagnosis. Procedia Computer Science. 2020;175:88–95. https://doi.org/10.1016/j.procs.2020.07.015

16. Fuan W, Hongkai J, Haidong S, et al. An adaptive deep convolutional neural network for rolling bearing fault diagnosis. Meas. Sci. Technol. 2017;28:095005.

17. Sohaib M, Kim J-M. Reliable fault diagnosis of rotary machine bearings using a stacked sparse autoencoder-based deep neural network. Shock Vib. 2018; 1–11. https://doi.org/10.1155/2018/1573482

18. Liu H, Yao D, Yang J, Li X. Lightweight convolutional neural network and its application in rolling bearing fault diagnosis under variable working conditions. Sensors (Basel). 2019;19(22):4827. https://doi.org/10.3390/s19224827 PMID: 31698734

19. Wang H, Liu Z, Peng D, Qin Y. Understanding and learning discriminant features based on multiattention 1DCNN for wheelset bearing fault diagnosis. IEEE Trans Ind Inf. 2020;16(9):5735–45. https://doi.org/10.1109/tii.2019.2955540

20. Jiby Mariya J, Jeeva J. Energy-reduced bio-inspired 1D-CNN for audio emotion recognition. Int J Sci Res Comput Sci Eng Inf Technol. 2025;11(3):1034–54. https://doi.org/10.32628/cseit25113386

21. Plakias S, Boutalis YS. Fault detection and identification of rolling element bearings with attentive dense CNN. Neurocomputing. 2020;405:208–17. https://doi.org/10.1016/j.neucom.2020.04.143

22. Zhao R, Yan R, Wang J, Mao K. Learning to monitor machine health with convolutional bi-directional LSTM networks. Sensors (Basel). 2017;17(2):273. https://doi.org/10.3390/s17020273 PMID: 28146106

23. Pan H, He X, Tang S, et al. An improved bearing fault diagnosis method using one-dimensional CNN and LSTM. Journal of Mechanical Engineering. 2018; 64(7/8): 443–52.

24. Wang Y, Deng L, Zheng L, Gao RX. Temporal convolutional network with soft thresholding and attention mechanism for machinery prognostics. Journal of Manufacturing Systems. 2021;60:512–26. https://doi.org/10.1016/j.jmsy.2021.07.008

25. Jiang M, Shao H. A CNN-transformer combined remote sensing imagery spatiotemporal fusion model. IEEE J Sel Top Appl Earth Observations Remote Sensing. 2024;17:13995–4009. https://doi.org/10.1109/jstars.2024.3435739

26. Wang D, Li M. Stochastic configuration networks: fundamentals and algorithms. IEEE Trans Cybern. 2017;47(10):3466–79. https://doi.org/10.1109/TCYB.2017.2734043 PMID: 28841561

27. Wang Q, Yang C, Ma X, et al. Underground airflow quantity modeling based on SCN. Acta Automatica Sinica. 2021; 47(8): 1963–75.

28. Wang Q, Dau W, Lu Q, et al. A sparse learning method for SCN soft measurement model. Control and Decision. 2022; 37(12): 3171–82. https://doi.org/10.13195/j.kzyjc.2021.1047

29. Liu J, Liu Y, Ma Y, et al. Smoothing L1 regularization for stochastic configuration networks. Control and Decision. 2024; 39(03): 813–8. https://doi.org/10.13195/j.kzyjc.2022.1612

30. Xu Z, Chang X, Xu F, Zhang H. L1/2 regularization: a thresholding representation theory and a fast solver. IEEE Trans Neural Netw Learn Syst. 2012;23(7):1013–27. https://doi.org/10.1109/TNNLS.2012.2197412 PMID: 24807129

31. Zheng X, Li J, Yang Q, Li C, Kuang S. Prediction method of mechanical state of high-voltage circuit breakers based on LSTM-SVM. Electric Power Systems Research. 2023;218:109224. https://doi.org/10.1016/j.epsr.2023.109224

32. Jin J, Xu Zi, Li C, et al. Research on rolling bearing fault diagnosis based on deep learningand support vector machine. Journal of Engineering for Thermal Energy and Power, 2022; 37(06): 176–84. https://doi.org/10.16146/j.cnki.rndlgc.2022.06.024

33. He M, Jiang W, Gu W. TriChronoNet: advancing electricity price prediction with multi-module fusion. Applied Energy. 2024;371:123626. https://doi.org/10.1016/j.apenergy.2024.123626

34. Lu Y, Wang W, Bai R, Zhou S, Garg L, Bashir AK, et al. Hyper-relational interaction modeling in multi-modal trajectory prediction for intelligent connected vehicles in smart cites. Information Fusion. 2025;114:102682. https://doi.org/10.1016/j.inffus.2024.102682

35. Yang B, Wang X, Xing Y, Cheng C, Jiang W, Feng Q. Modality fusion vision transformer for hyperspectral and LiDAR data collaborative classification. IEEE J Sel Top Appl Earth Observations Remote Sensing. 2024;17:17052–65. https://doi.org/10.1109/jstars.2024.3415729

36. Jiang W, Zhang Y, Han H, Huang Z, Li Q, Mu J. Mobile traffic prediction in consumer applications: a multimodal deep learning approach. IEEE Trans Consumer Electron. 2024;70(1):3425–35. https://doi.org/10.1109/tce.2024.3361037