Class-imbalanced flow meter fault diagnosis under small samples using reinforcement learning based Mahalanobis Taguchi system

Abstract

Flow meter is one of the most essential sensors in industrial development, energy measurement and environmental protection. Monitoring of flow meter performance can help detect anomalies early and enable timely corrective actions for critical industrial equipment in harsh operating environments. However, flow meter diagnostic models are often prone to overfitting and low accuracy caused by class-imbalanced small-sample data. To address these problems, a reinforcement learning Mahalanobis Taguchi system (RLMTS) model is proposed in this paper, which primarily consists of three modules, namely Mahalanobis space (MS) construction, threshold determination, and sample classification. In the MS module, an initial MS is constructed by selecting variables through orthogonal array design and signal-to-noise ratio analysis. Reinforcement learning is then introduced to adaptively refine the MS which is verified by the Mahalanobis distance. In the threshold determination module, a neural network algorithm is proposed to replace the traditional quality loss function for optimal threshold determination. In the sample classification module, the fault diagnosis of unknown samples is performed using the valid MS and calculated Mahalanobis distance. Experimental results show that the proposed RLMTS is not only suitable for flow meter fault diagnosis under different class-imbalance ratios with different small sample sizes, but also demonstrates a better diagnostic performance, stronger robustness, and broader applicability compared to the 19 benchmark diagnosis models. The use of RLMTS therefore guarantees stable operation of the flow meters, contributing to energy savings and environmental protection.

Keywords

Flow meter diagnosis; Class-imbalanced small-sample data; Mahalanobis Taguchi system; Reinforcement learning; Neural network algorithm

1. Introduction

Flow meter is an essential part of the equipment that effectively measures the total amount of fluid over a selected time interval. It is widely used in industrial development [1], energy measurement [2] and environmental protection [3]. A flow meter failure can lead to severe consequences, including: (i) measurement errors, resulting in the reduced measurement accuracy; (ii) slower equipment response, reducing operational stability, reliability, and efficiency; (iii) energy waste and financial losses; and (iv) environmental pollution and safety risks. Therefore, real-time condition monitoring and fault diagnosis of flow meters are of substantial practical significance.

In the related research on the fault diagnosis of flow meters, Chen et al. applied kernel principal component analysis (PCA) to reduce the data dimensionality, and used the improved support vector machine (SVM) to realize the fault diagnosis of the flow meter [4]. Oh et al. employed fault-tolerant control and standards to diagnose faults in thermocouples or mass flow meters [5]. Liu et al. utilized a backpropagation neural network based on the Adam optimization algorithm to detect the fault of the flow meters in the air conditioning water system [6]. Qiu et al. proposed an auxiliary alarm rule to solve the trade-off between sensitivity and false alarm rate for fault detection and diagnosis of flow meters [7]. However, research on flow meter fault diagnosis overall remains limited, and most existing studies are based on big data.

In practice, due to harsh operating conditions, data collected from flow meters in large, high-cost equipment often exhibits class imbalance and small sample sizes[8]. The class-imbalanced small-sample data problem refers to the uneven distribution of response variable values [9] and limited sample size [10], posing a challenge for flow meter fault diagnosis. This can lead to a general fact that the higher the imbalance ratio (IR) of the data, the worse the diagnostic performance of the diagnostic models [11]. This also makes the diagnostic model prone to overfitting, high variance and poor generalization ability [12–14].

Current research on equipment fault diagnosis under class-imbalanced small sample data primarily focuses on increasing sample size. Zhang et al. enhanced the downsampling module by using density-based spatial clustering of applications with noise and distribution ratio calculation, then integrated it into a Siamese network. By combining the Siamese network with data augmentation techniques, the diagnostic accuracy of the model was improved [15]. Fan et al. used a resampling technique to

extend the number of small samples, combined conditional labeling in a conditional generative adversarial network (GAN) with a Wasserstein GAN and gradient penalties to control the generated samples and filter the samples using the maximum mean difference, and ultimately utilized a convolutional neural network (CNN) to achieve fault diagnosis [16]. Zhang et al. first incorporated the dual-stream adaptive deep residual shrinkage block into the Vision Transformer architecture to adaptively remove redundant information while preserving critical local fault features, and then applied an interclass-intraclass rebalancing loss function to optimize model convergence [17]. Li et al. utilized Wasserstein GAN to achieve data balancing, and then used long short-term memory (LSTM) full convolutional network to realize fault diagnosis in a small-sample working environment [18]. Shen et al. generated high-quality pseudo-samples from limited faulty data, employed an improved Wasserstein GAN with gradient penalty to balance the number of faulty and normal samples, and applied a LSTM network for fault diagnosis [19]. Wang et al. integrated the improved majority weighted minority oversampling technique and least squares SVM to develop an interpretable and optimizable fault diagnosis model [20]. Although these methods offer effective solutions, the process of generating new samples inevitably introduces noise, which may degrade sample quality and diagnostic model performance.

In this context, the Mahalanobis Taguchi system (MTS) emerges as a promising approach. This method constructs a Mahalanobis space (MS) using a single-class sample, offering distinct advantages for handling class-imbalanced data [21]. In addition, MTS requires fewer training samples than shallow machine learning models and deep learning networks. However, the MTS emphasizes the use of orthogonal array to arrange variables and reduce the number of trials; the use of orthogonal array and signal-to-noise ratio (SNR) gain to obtain the MS may not necessarily be an optimal solution [22]. Meanwhile, the MTS uses the quality loss function in determining the classification boundaries, whose parameters are determined by the decision cost, which is not easy to obtain from the relevant data [23]. These issues need to be addressed when dealing with MTS.

Reinforcement learning (RL) supports balanced exploration of new feature combinations versus utilizing known valid combinations, and has been widely used in the field of fault diagnosis. Fan et al. proposed a novel, general imbalanced sample selection strategy based on deep RL to address industrial equipment fault diagnosis in imbalanced scenarios [24]. Li et al. used an advantage actor-critic algorithm to extract important features and combined it with synthetic minority over-sampling technique (SMOTE) to process imbalanced data for bearing fault diagnosis [25]. Zhu et al. employed the deterministic learning theory to pre-train neural networks and subsequently integrated RL to adaptively optimize network parameters, thereby enhancing the model's robustness. This approach ultimately enabled effective fault diagnosis for rotating machinery [26]. Zhang et al. incorporated policy gradient and actor-critic strategies into a cost-sensitive classifier to achieve fault diagnosis in imbalanced scenarios [27].

Thus, a reinforcement learning Mahalanobis Taguchi system (RLMTS) is proposed in our study for flow meter fault diagnosis under class-imbalanced small-sample data. First, the advantages of MTS are fully utilized to construct the MS with a single class of samples as a benchmark to process the imbalanced data. Second, RL is introduced to optimize the MS. The features with SNR gain greater than zero are used as the initial action space. The negative value of the loss function is taken as the reward, and the goal is to maximize the cumulative reward. Based on the given exploration strategy, the MS is iteratively optimized. Finally, a neural network algorithm (referred to as NNA in our paper) is employed to determine and optimize the threshold of traditional MTS, thus realizing an effective sample classification for fault diagnosis. NNA is not a traditional neural network model, instead, it is a nature-inspired metaheuristic optimization method that mimics the dynamic information propagation behavior of biological neural networks. Through population-based exploration, the algorithm evaluates the current best solution and its corresponding objective value, and iteratively updates them based on the minimization of a predefined objective function. The final solution represents the optimal threshold.

Compared with existing methods, RLMTS does not rely on data augmentation techniques, thereby reducing the risk of introducing noise and enhancing its reliability in diagnostic tasks. Moreover, it does not require the construction of sample pairs, which lowers the dependency on data preprocessing and makes it more flexible and broadly applicable across different fields.

The main contributions of this paper are summarized as follows.

(i) A systematic diagnostic framework: a new diagnostic framework is proposed for class-imbalanced

flow meter fault diagnosis under small samples, including data preprocessing, MS construction, threshold determination and sample classification. Experiments show that the proposed framework can maximize the difference in Mahalanobis distances (MDs) between normal and abnormal samples to handle fault diagnosis under different IRs with different small sample sizes, achieving great diagnostic performance.

(ii) An effective feature selection strategy: the variables selected by the orthogonal array and SNR gain are used as the initial MS, which is then continuously optimized through the interaction between the agents and environment, while considering the feature selection rules to explore a better feature set.

(iii) An efficient way to determine classification boundaries: introduce NNA to leverage its accuracy, computational efficiency and adaptability, overcoming the subjectivity of the quality loss function and obtaining better boundary lines for samples of different classes.

(iv) Theoretical innovation and application value: the ablation study and comparative analysis with the benchmark models demonstrate that the proposed RLMTS provides a theoretical innovation to the MTS with a better diagnostic accuracy and expands the application field of MTS. RLMTS can effectively realize the fault diagnosis of class-imbalanced small-samples, which is conducive to reducing the measurement error, improving the operational efficiency of the equipment, ensuring the reliability and stability of equipment operation, and thus preventing the occurrence of safety accidents and reducing economic losses. Furthermore, due to its adaptability to numerical features, RLMTS exhibits generalizability to other sensor fault diagnosis tasks, such as temperature and pressure sensors. This highlights its potential for broader applications in industrial fault detection and diagnosis across diverse sensor-based systems.

The rest of the paper is organized as follows. Section 2 presents the fundamental theories of MTS and RL. Section 3 describes the RLMTS in detail. Section 4 first verifies the feasibility of RLMTS via ablation studies, evaluations under different imbalance ratios and sample sizes, and an analysis of the impact of data characteristics. It then demonstrates the effectiveness of RLMTS through three case studies. Finally, the performance of all diagnostic models is analyzed and discussed in detail, with particular attention to the reasons behind their strengths and limitations across different datasets. Section 5 presents the conclusion.

2.Fundmendal theories

2.1. Mahalanobis Taguchi system

MTS, proposed by Japanese quality expert Taguchi [28], mainly consists of MD and Taguchi method. MD is a distance metric that extends and refines the Euclidean Distance. MD eliminates the effect of magnitude and can cope with the non-independent homogeneous distribution among dimensions in dataset with high-dimensional linear distribution, which can be used for handling ellipsoidal data [29].

The original data (sample size is n and the number of variables is m) is normalized to obtain $X_{n\times m}$, where x_i is the ith sample of $X_{n\times m}$, $i=1,2,\cdots,n$. The MDs of the samples can be calculated.

$$MD_i = x_i R^{-1}(x_i)^T \tag{1}$$

where R is the matrix of correlation coefficients.

For the Taguchi method, the orthogonal array and the SNR are two important components. Taguchi method is a balanced design of all control parameters, setting the optimal combination of different control parameters from their responses, and ultimately achieving the goal of maximum efficiency with a minimum number of trials [30].

The SNR is calculated using an orthogonal array $L_g(m^q)$, where g denotes the number of trials and q denotes the level of factors. In MTS, $q=2$, i.e., each variable has two levels, representing "select this variable" or "do not select this variable". The higher the SNR, the better the selection of the feature subset. SNR is calculated by eq. (2) as below.

$$SNR = 10\log\frac{1}{n_1}\sum_{i=1}^{n_1} x_i^2 \tag{2}$$

where n_1 is the number of abnormal samples. Note that n_1 depends on the training data rather than the orthogonal array.

2.2. Reinforcement Learning

RL serves as a powerful learning framework capable of learning complex strategies in high-dimensional environments, having sample efficient yet sensitive to hyperparameters settings, and balancing the relationship between existing information and environment exploration [31]. Fig. 1 illustrates the RL process schematically used in our study.

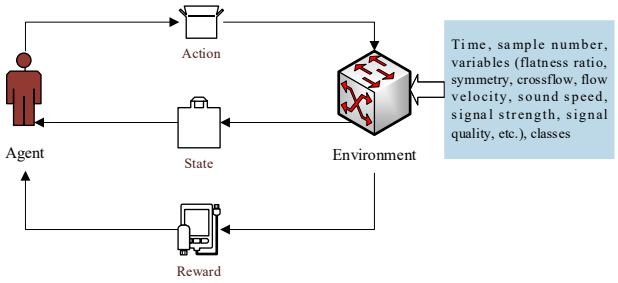


Fig. 1. The structure of the RL process.

The RL framework is often modeled as a Markov decision process with an infinite temporal horizon. A Markov decision process consists of three key components: a set of possible states, a set of actions, and immediate rewards. In our study, the states encompass all environmental variables directly influenced by the agent, including time, sample number, flow meter variables such as flatness ratio, symmetry, crossflow, flow velocity, sound speed, signal strength and signal quality, and class attributes. In RL, the agent observes the environment, determines an action based on the current state, and receives a reward after interacting with the environment. The agent iteratively learns to maximize expected rewards or achieve other optimization objectives.

3. Methodology

3.1. Framework overview

Fig.2 presents an overview of the RLMTS framework.

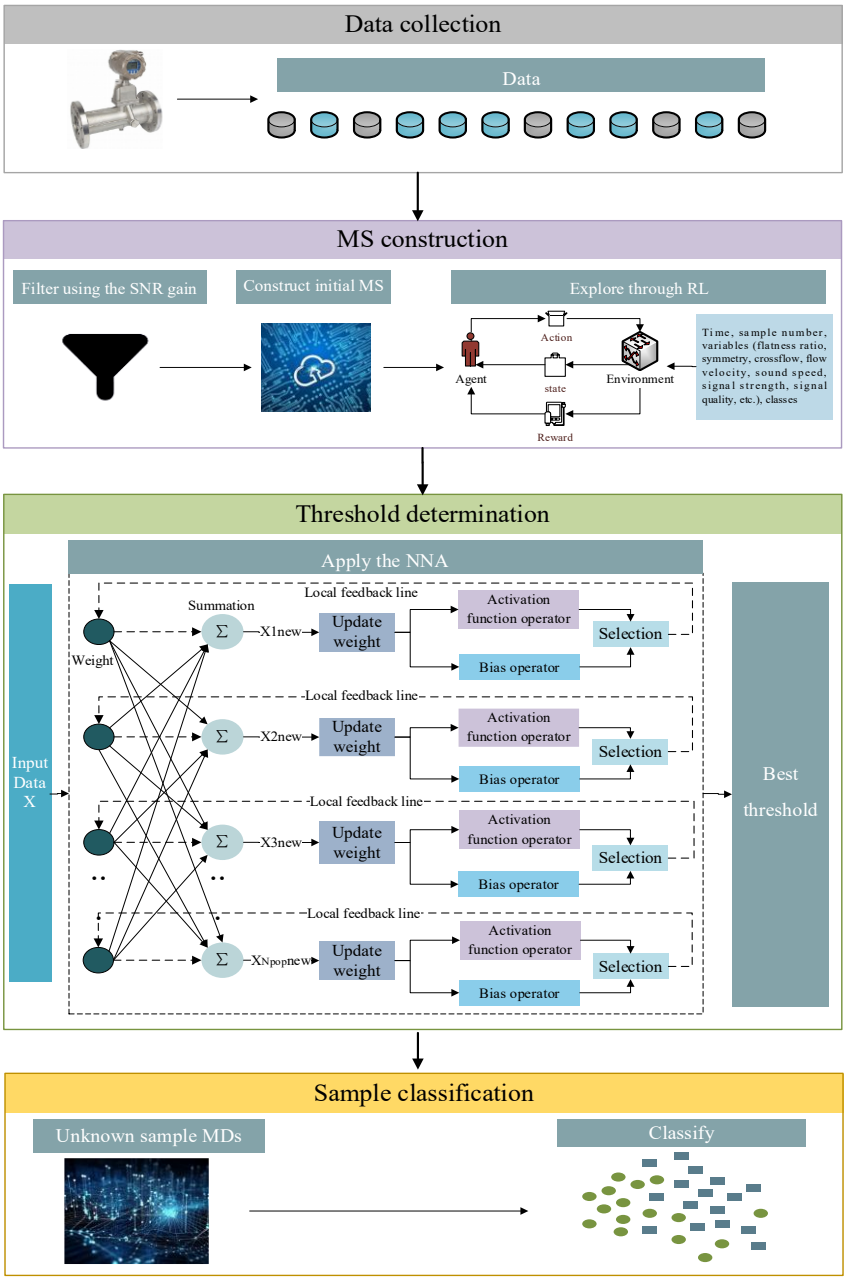


Fig. 2. Overview of the RLMTS framework.

RLMTS consists of three modules: MS construction, threshold determination, and sample classification. The MS construction module initializes the MS using variables filtered by SNR gain, and then leverages RL to optimize

the MS composition, ensuring the retention of essential information while eliminating redundancy. This module lays an important foundation for accurate sample classification. The threshold determination module aims at minimizing the fitness function and employs the NNA to complete exploration of the optimal threshold, which is an indispensable step for achieving accurate sample classification. The sample classification module effectively classifies unknown samples and identifies faulty ones. The following sections provide a detailed description of each module.

3.2. Mahalanobis space construction

The MD and SNR of the input samples are calculated according to eqs. (1) and (2). Note that in order to obtain a more robust MD, the inverse matrix is replaced by a Moore-Penrose pseudoinverse matrix in eq. (1).

Suppose SNR_1 is the SNR with the variable selected and SNR_2 is the SNR without the variable selected. Then the SNR gain is calculated as below.

$$SNR_{gain} = SNR_1 - SNR_2 \tag{3}$$

Since a higher SNR indicates better class separability, a feature should be considered important for construction of the MTS if its inclusion leads to an increase in the overall SNR. In other words, if the SNR gain obtained by selecting a feature is greater than zero, the feature is deemed valuable. Therefore, features with an SNR gain greater than zero are selected to form a subset of features. The MDs of the normal and abnormal samples are then recalculated and compared. If the MD of the abnormal samples exceeds that of the normal samples, the initial MS is considered valid.

If the MS is invalid, it is necessary to reconstruct a new MS. For the variables with a SNR gain greater than zero, the variables corresponding to smaller values are removed, while for the variables with a SNR gain less than zero, the variables corresponding to larger values are added to construct a new MS. After the new MS is constructed, the validity of the new MS needs to be verified.

We have used the RL to optimize the MS composition. Given state $s_t \in S$ and action $a_t \in A$ at the time moment t, let $r_t(s_t, a_t) \in R$ be the reward, $\gamma \in [0,1)$ be the discount factor (constant), V^* be the optimal state value function, and s_{t+1} be the state at the next time moment $(t+1)$. The optimal action-value function Q^* is calculated by eq. (4).

$$Q_t^*(s_t, a_t) = r_t(s_t, a_t) + \gamma \sum_{s_{t+1}} P(s_{t+1} | s_t, a_t) V^*(s_{t+1}) \tag{4}$$

where P denotes the probability.

The optimal policy π^* can be obtained from the Q^*.

$$\pi_t^*(s_t) = \arg\max_{a_t \in A} Q^*(s_t, a_t) \tag{5}$$

At this point, the action taken by the agent each time is determined by eq. (5). This means that the agent executes a greedy strategy by which each action is made based on the current experience and the principle of maximizing subsequent benefits.

In order to realize the exploration of the environment by the agent, the ε greedy strategy is introduced. The agent randomly selects an action with a probability of ε and executes the greedy strategy with a probability of (1-ε). ε greedy strategy is chosen due to its simplicity, ease of implementation, and proven effectiveness in balancing exploration and exploitation in discrete action spaces. It provides a stable convergence in environments with limited action complexity, such as feature selection tasks.

The goal of action exploration is to maximize the value of action-value function Q. When the action is executed, the feature subspace in the MS changes, and the Q-value changes accordingly. The Q-value update is shown in eq. (6).

$$\begin{aligned} Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \lambda [r(s_t, a_t) \\ + \gamma \max_a Q(s_{t+1}, a_t) - Q(s_t, a_t)] \end{aligned} \tag{6}$$

where $\lambda \in (0,1]$ is the step size.

For each iteration, the agent interacts with the environment (feature subset) and performs ε greedy strategy, $\varepsilon \in (0,1)$. Let $c \in (0,1)$ be a randomly generated constant.

$$\begin{cases} a_t^*(s_t, a_t) = r(s_t, a_t) + \gamma \sum P(s_{t+1} | s_t, a_t) V^*(s_{t+1}), & c < \varepsilon \\ 0 \ \textit{or}\ 1, & \textit{others} \end{cases} \tag{7}$$

Meanwhile, the feature selection and NNA (to be described in the subsequent section) are wrapped up and processed during agent exploration. The action rules are therefore formulated.

Let F be the fitness function and F_t^* be the optimal fitness function at the time moment t.

$$F = 2 \times count(MD_x > K) + count(MD_y < K) \tag{8}$$

where MD_x and MD_y represent the MDs of normal and abnormal samples, respectively, $count(\cdot)$ refers to the number of eligible samples to be counted, and K is the current optimal solution obtained.

Let D be the number of feature selection cases in actions a_{t+1} and a_t that differ. For example, action a_t contains the variables number 1, 2, 3, 4, 5, 6, while a_{t+1} contains the variables number 1, 2, 5, 6, meaning $D = 2$.

When $D = 1$,

$$a = \begin{cases} a_{t+1}, F_{t+1}^* > F_t^* \\ a_t, F_{t+1}^* \le F_t^* \end{cases} \tag{9}$$

When $D > 1$, the agent needs to fix the choice of any two variables in a_t during the exploration. Assume a_{t_new}

and a_{t+1_new} as the actions made by the agent after fixing any two feature selection cases at time moments t and $(t+1)$, respectively.

$$a=\begin{cases} a_{t+1\_new}, F_{t+1}^{*} > F_{t}^{*} \\ a_{t\_new}, F_{t+1}^{*} \le F_{t}^{*} \end{cases} \tag{10}$$

After the iterations are completed, the MDs of the normal and abnormal samples are recalculated according to the actions given by the agent. If the MD of the normal samples is lower than that of the abnormal samples, the MS construction is valid, otherwise it is necessary to return to eq. (7) and the exploration process repeats again.

3.3. Threshold determination

To better determine the threshold, NNA is introduced into the MTS. The NNA is not a conventional feedforward neural network with explicit layers, neurons, or activation functions. Instead, it is a nature-inspired metaheuristic optimization method proposed by Sadollah et al. [32], which simulates the dynamic information propagation mechanism of biological neural systems. By mimicking the way connection weights are transmitted among neurons, NNA searches for global optima in continuous spaces. Unlike traditional neural networks, NNA does not possess a layered structure, making it simpler to implement and computationally efficient. The core idea of NNA is to emulate the principle of biological evolution and group intelligence, and transform the optimization problem into a process of searching for the optimal solution.

The NNA is adopted to adaptively determine the optimal classification threshold by minimizing eq. (8), which defines the misclassification cost based on MDs. The algorithm begins by initializing a population and evaluating the initial fitness of each candidate solution. It then simulates the propagation of neural connections through a weight matrix to generate new solutions, by incorporating perturbations to enhance exploration. During each iteration, the search range is progressively narrowed to promote convergence; the objective function is recalculated; and the historical best solution is retained. This global search mechanism enables NNA to effectively avoid local optima and ensures reliable threshold optimization for the RLMTS model.

Detailed steps are described as follows.

3.3.1. Initialization

Set N_{pop} as the number of populations, T as the number of iterations, $T_{\max}$ as the maximum number of iterations, $T=1,2,\cdots,T_{\max}$, w_0 as initial weight matrix, and w as a square matrix of uniformly generated random numbers between 0 and 1 of size $N_{pop}\times N_{pop}$.

$$w=\begin{bmatrix} w_{11}, w_{21}, \cdots, w_{N_{pop}1} \\ w_{12}, w_{22}, \cdots, w_{N_{pop}2} \\ \vdots \\ w_{1N_{pop}}, w_{2N_{pop}}, \cdots, w_{N_{pop}N_{pop}} \end{bmatrix} \tag{11}$$

The eq. (11) satisfies, $\sum_{j=1}^{N_{pop}} w_{ij}=1$, $w_{ij}\in(0,1)$, $i,j=1,2,\cdots,N_{pop}$, w_{ij} denotes the ith row and jth column element of the weight matrix w. The initial weight matrix w_0 is calculated in Algorithm 1.

Algorithm 1 Initialize w_0

Require: N_{pop}

W_0 <-- a diagonal matrix of size $N_{pop}\times N_{pop}$ with all diagonal values 0.5

For i from 1 to N_{pop} do:

t = Random(1, N_{pop}-1)×0.5, where Random(a, b) refers to randomly generating a random array of size a×b.

t = t / Sum(t)×0.5, where Sum(.) is a summation formula

Elements in w_0 with a value of zero<-- values in t

End For

3.3.2. Weight matrix, bias operator and pattern solution update

We randomly generates $r_1\in(0,1)$ to update the weight matrix w, and also randomly select a number of columns in r_1 and replace these columns in w with the columns in r_1, then normalize it.

To fast obtain the optimal solution, NNA searches for the solution within the restricted search space. Set the lower limit of the search space as l, the upper limit of the search space as u, the position of the Tth iteration as p_T (initial solution is p_0) and the pattern solution of the Tth iteration as K_T (initial pattern solution is K_0).

$$\begin{cases} p_0 = r_2\times(u-l)+l \\ K_0 = p_0 \end{cases} \tag{12}$$

where $r_2\in(0,1)$ is a random number.

The p_T is then updated and the following steps as given by eq. (13) are iterated over until the predefined iteration stopping condition is satisfied or $T=T_{max}$.

$$\begin{aligned} p_T &= w_T \times p_{T-1} \\ p_T &= p_T + p_{T-1} \end{aligned} \tag{13}$$

The weight matrix can also be updated.

$$\begin{aligned} w_j &= \left| w_j + 2\times r_3 \times w^e - w_j \right| \\ w_j &= w_j / sum(w_j) \end{aligned} \tag{14}$$

where w^e is the target weight and $r_3\in(0,1)$ is a random number.

The bias operator β acts as noise and the initial value $\beta_0 = 1$. Two methods can be used for updating the bias operator β, as given in eq. (15) by Sadollah et al. [32].

$$\begin{aligned} \beta_{T+1} &= \beta_T \times 0.99 \\ \beta_{T+1} &= 1 - T/T_{\max} \end{aligned} \tag{15}$$

In order to minimize the effect of the manually set parameter $T_{\max}$ on β, we choose method $\beta_{T+1} = \beta \times 0.99$ here.

The pattern solution K_T can be updated as below.

$$K_T = K_T + 2 \times r_4 \times (F_T^* - K_T) \tag{16}$$

where $r_4 \in (0,1)$ is a random number.

Finally, the best solution among all K_T is obtained, as denoted as K^*, and the optimal threshold is therefore determined.

3.4. Sample classification

According to the MS constructed in Section 3.2, the MDs of the samples to be tested (MD_Z) is calculated. The class of the samples is determined by comparing between the MD_Z and K^* obtained from the calculation in Section 3.3.

$$\begin{cases} faulty, K^* < MD_z \\ normal, others \end{cases} \tag{17}$$

To effectively assess the performance of the proposed diagnostic model, evaluation is conducted using the confusion matrix. In addition, precision, recall, specificity, accuracy, F1-score, and G-mean are also adopted as key performance metrics [33].

The proposed RLMTS is presented in Fig. 3. The methodology of RLMTS is demonstrated in Algorithm 2.

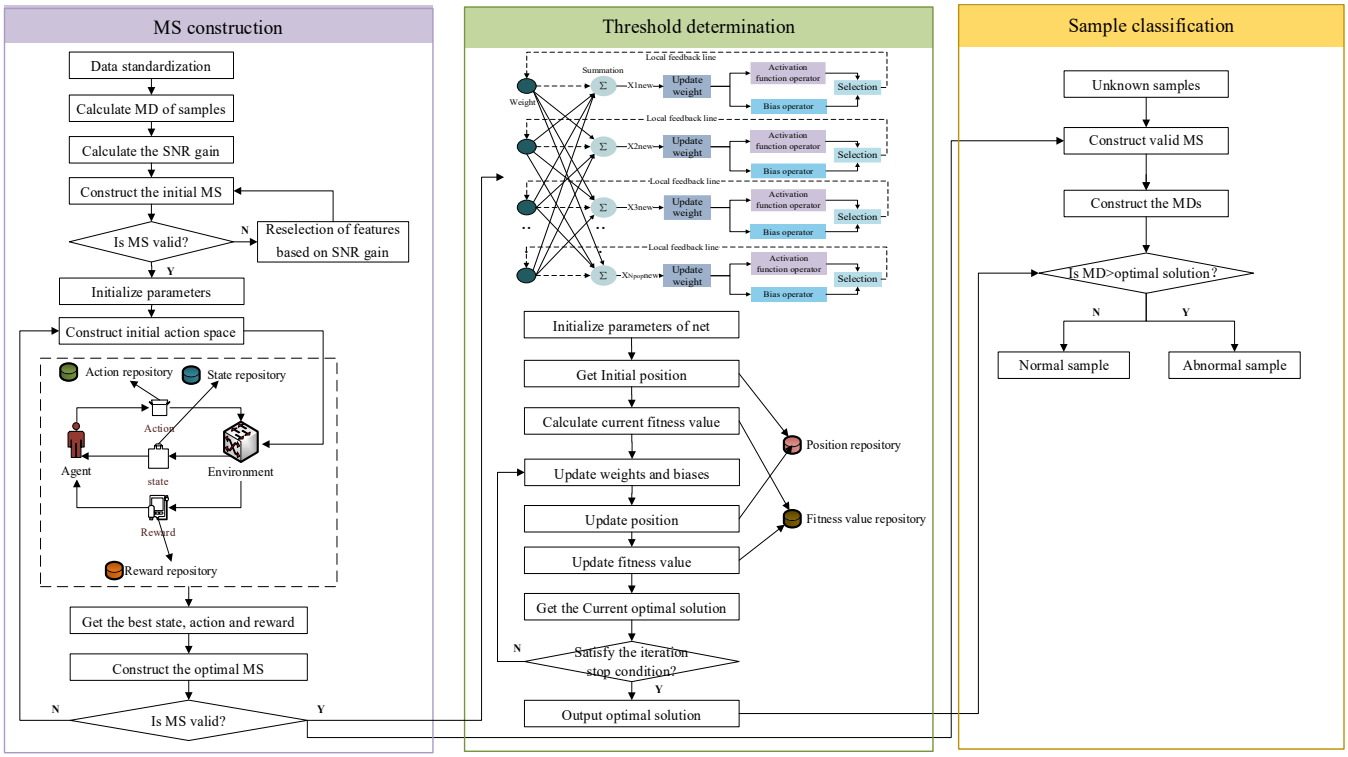


Fig. 3. The RLMTS.

Algorithm 2 RLMTS

Data standardization

Calculate the sample MD according to eq. (1)

Calculate the signal-to-noise gain according to eqs. (2)-(3) to construct the initial MS

a_0 <-- initial MS

Interact with the environment to get the reward r_0

For t=1 to maximum number of moments, do:

Agent adopts **ε** greedy strategy and obtains the action strategy based on eqs. (4)-(7)

Update rewards according to eq. (8), note that r and F are inversely proportional.

When $D=1$, do:

If $F_{t+1}^{opt} > F_t^{opt}$, do:

$a=a_{t+1}$

Else do:

$a=a_t$

End if

When $D>1$, do:

Fix the choice of any two variables in a_t

If $F_{t+1}^{opt} > F_t^{opt}$, do:

$a=a_{t+1\_new}$

Else do:

a=a_{t_new}
End if
Recalculate the MD to test the validity of MS
For T=1 to maximum number of iterations, do:
Calculate the initial p_0, K_0 and w_0 according to eqs. (11)-(12) and Algorithm 1
Update the position and optimal solution based on eq. (13)
Based on eq. (14) update the weight matrix
Update bias according to eq. (15)
Compare and save the optimal solutions from the iterations and get the K^*
End for
End for
Calculate the MD of the unknown sample based on the optimal MS

If $MD_z > K$, do:
Output ('Faulty sample')
Else do:
Output ('Normal sample')
End if

4. Experimental results and discussion

4.1. Datasets

The three datasets used with different IRs were taken from Gyamfi et al. [34]. These datasets were collected from real ultrasonic flow meters: Dataset A was acquired from an 8-path ultrasonic flow meters, while Datasets B and C were obtained from 4-path ultrasonic flow meters. Table 1 provides a detailed description of the datasets. Table 2 illustrates the physical meaning of the variables in the datasets.

Table 1
Description of the datasets.

Dataset	Sample size	Number of variables	Majority class sample sizes	Minority class sample sizes	IR
A	87	37	52	35	1:1.49
B	181	44	127	54	1:2.35
C	180	44	129	51	1:2.53

Table 2
Physical meaning of variables in datasets.

Dataset	Variable No.	Variable meaning	Variable No.	Variable meaning
Dataset A	1	Flatness ratio	2	Symmetry
	3	Crossflow	4-11	Flow velocity in each of the eight paths
	12-19	Speed of sound in each of the eight paths	20	Average speed of sound in all eight paths
	21-36	Gain at both ends of each of the eight paths	37	Class attribute
Dataset B & C	1	Profile factor	2	Symmetry
	3	Crossflow	4-7	Flow velocity in each of the four paths
	8-11	Speed of sound in each of the four paths	12-19	Signal strength at both ends of each of the four paths
	20-27	Signal quality at both ends of each of the four paths	28-35	Gain at both ends of each of the four paths
	36-43	Transit time at both ends of each of the four paths	44	Class attribute

4.2. Parameter design

All experiments are implemented using Python 3.10.12 and Matlab R2022a on a system equipped with a 12th Gen Intel® Core™ i7-12700H processor (up to 3.69 GHz), 16 GB of DDR5 RAM (4800 MT/s), and an NVIDIA GeForce RTX 3060 Laptop GPU (6 GB VRAM).

In RLMTS, the environment includes time, samples,

and features, etc. State is the set of currently selected features. Action is a modification operation on the feature space. Reward is the opposite of the loss function, i.e., the opposite of eq. (8). Learning rate is 0.1, discount factor $\gamma = 0.95$, $\varepsilon = 0.9$ and the number of iterations is 100.

In our study, we set population size $N_{pop} = 100$. Table 3 gives details of the search space, where ξ-fold means the ξth fold in the five-fold cross-validation, l is the lower limit of the search space, while u is the upper limit of the search space.

Table 3
Search space of the datasets.

Dataset	ξ-fold	l	u
Dataset A	1	0.97	10.17
	2	0.96	12.35
	3	0.97	31.03
	4	0.96	21.63
	5	0.96	32.48
Dataset B	1	0.98	7572.22
	2	0.98	8164.96
	3	0.98	7722.28
	4	0.98	8653.65
	5	0.97	10008.93
Dataset C	1	0.98	2006.74
	2	0.98	1867.84
	3	0.98	1937.41
	4	0.97	1907.46
	5	0.97	2388.65

4.3. Ablation Study

4.3.1. Validity of RL and NNA

To validate the effectiveness of RLMTS and assess the significance of its core components, ablation studies are conducted through sequential removal of the RL module and NNA module from the complete framework, resulting in two simplified architectures: Model 1 (RLMTS without RL) and Model 2 (RLMTS without NNA). Five-fold cross-validation is implemented to comprehensively evaluate model performance and generalization capabilities. This validation methodology divides the original dataset into five equal-sized subsets (folds), with four subsets designated for model training and the remaining subset reserved for testing. Table 4 presents the mean values of the evaluation metrics after the ablation experiments are performed on three datasets, and each dataset is subjected to the five-fold cross-validation.

As can be seen from Table 4, the model diagnostic performance is not robust after excluding the RL module or the NNA module; therefore, both the RL and the NNA are crucial parts for the RLMTS model.

Table 4
Results of the ablation study.

Models	Accuracy	F1	G-mean
RLMTS	0.99	0.98	0.99
Model 1	0.93	0.86	0.95
Model 2	0.60	0.61	0.66

4.3.2. Validity of SNR gain

In order to verify its effectiveness, the SNR gain method is compared with two feature selection methods, mutual information (MI) and PCA. Since a greater MD difference between normal and abnormal samples makes classification easier and achieves higher accuracy, the MD difference is used as an evaluation index. Table 5 shows the results of the three feature selection methods in detail.

Table 5
Sample MD differences using SNR gain, MI and PCA.

Dataset	Models	MD difference mean	MD difference standard deviation
A	SNR gain	294.87	221.92
	MI	200.72	64.50
	PCA	21.67	26.60
B	SNR gain	30850.97	5206.33
	MI	1572.04	220.47
	PCA	17.45	20.31
C	SNR gain	6221.45	1036.63
	MI	178.81	57.58
	PCA	848.37	693.24

As can be seen from Table 5, the MD difference between normal and abnormal samples computed using SNR gain is the most pronounced, which is, therefore, more favorable for the model to perform the classification task.

4.3.3. Validity of ε-greedy strategy

To verify its effectiveness, the ε-greedy strategy is compared with Thompson sampling. Using the two strategies, the values of the evaluation metrics are calculated in conjunction with other modules of RLMTS. Fig. 4 shows the results of the model, with metric values averaged across three datasets, each of which is evaluated using five-fold cross-validation.

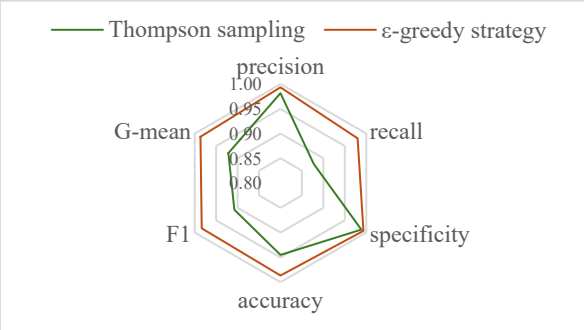


Fig.4. Comparison of results from ε-greedy and Thompson sampling.

As can be seen in Fig. 4, the combination of ε-greedy strategy with other modules of RMLTS has better diagnostic performance with higher values of all the evaluation metrics.

4.4. RLMTS for different IRs with different small sample sizes

4.4.1. Experiments with IR from 1:1.19 to 1:1.53

In order to investigate the diagnostic performance of RLMTS on small samples with different IRs, new datasets are created by taking 20%, 40%, and 60% of the data from datasets A, B, and C, respectively. These new datasets are named A-1, A-2, A-3, B-1, B-2, B-3, C-1, C-2, and C-3. RLMTS is then applied to the new datasets to conduct fault diagnosis. Table 6 gives the confusion matrix, demonstrating the diagnostic performance of RLMTS on the nine new datasets, where TP denotes true positives, TN denotes true negatives, FP denotes false positives, and FN denotes false negatives.

Table 6

Confusion matrix of RLMTS on the nine new datasets.

Dataset	IR	Feature number	TP	FN	FP	TN	Time
A-1			1	0	0	2	4.66
A-2	1:1.49	37	2	1	0	4	4.64
A-3			4	0	0	6	4.76
B-1			2	0	0	5	4.76
B-2	1:2.35	44	4	0	0	10	4.72
B-3			6	0	0	15	4.81
C-1			2	0	0	5	4.83
C-2	1:2.53	44	4	0	1	9	4.70
C-3			6	0	0	15	4.75

As can be seen from Table 6, except for one sample misjudged by RLMTS in Datasets A-2 and C-2, the rest samples are judged correctly, indicating that RLMTS has a great diagnostic capability.

4.4.2. Experiments with IR from 1:1.10 to 1:1.100

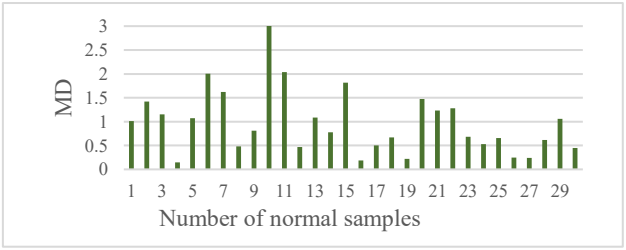
Due to the limited sample size of datasets A-C, it is difficult to explore the diagnostic performance of the RLMTS under higher IRs. Therefore, Dataset D [35] is employed, as an example, to investigate the diagnostic performance of the proposed RLMTS method under higher imbalance ratios ranging from 1:10 to 1:100. In the experiments for Dataset D, the minority class has 30 training samples and 10 test samples, with the same IR maintained in both training and testing. Table 7 presents in detail the RLMTS confusion matrix for higher imbalance ratios.

Table 7

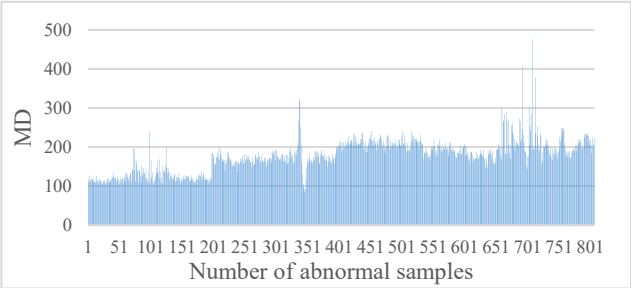
Confusion matrix of RLMTS on the Dataset D.

IR	TP	FP	FN	TN
1:10	10	0	0	100
1:20	10	0	0	200
1:30	10	0	0	300
1:40	10	0	0	400
1:50	10	0	0	500
1:60	10	0	0	600
1:70	10	0	0	700
1:80	10	0	0	800
1:90	10	0	0	900
1:100	10	0	0	1000

When IR=1:100, the number of training samples is 3030 (number of normal samples is 30 and that of abnormal samples is 3000) and the number of test samples is 1010 (number of normal samples is 10 and that of abnormal samples is 1000), and the optimal threshold obtained is 53.76. Fig. 5 shows the MDs of normal and part of the abnormal samples due to large abnormal samples in this case. It can be seen that there is a large difference between the MDs of the two types of samples, which illustrates that the RLMTS is still effective in identifying faulty samples even when the IR is much higher.



(a) MD of normal samples



(b) MD of abnormal samples

Fig.5. MDs of normal and abnormal samples under IR=1:100.

Consequentially, RLMTS is suitable for small-sample fault diagnosis with good diagnostic performance under different IRs.

4.4.3. Experiments with IR=1:1000

Due to the difficulty of data collection in real scenarios with IR=1:1000, the experiments are conducted using data generated based on different distributions.

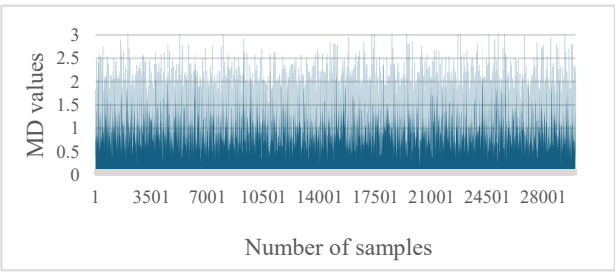
All samples have 25 feature dimensions. The training set contains 30,000 normal samples and 30 abnormal samples, while the test set includes 10,000 normal samples and 10 abnormal samples. The normal samples follow a multivariate standard normal distribution, whereas the abnormal samples are generated from a mixture of two abnormal Gaussian modes, with additional heavy-tailed perturbations introduced by a t-distribution, as described below.

$$x_{normal} \sim N(0, I_{25}) \tag{18}$$

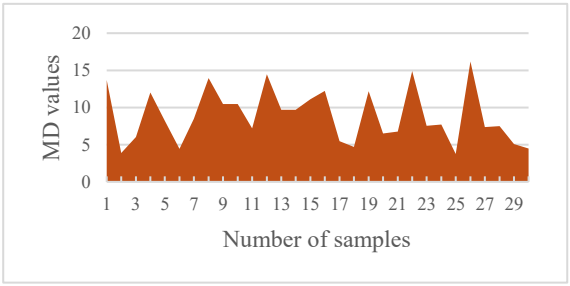
$$x_{anomaly} \sim \begin{cases} N(3, 2I_{25}) + 0.5t_3 \\ N(-2, 1.5I_{25}) + 0.5t_3 \end{cases} \tag{19}$$

The features with SNR gain greater than zero correspond to indices 1 through 23. After RL exploration, the selected feature indices are 2, 9, 11, 12, 16, 18, 19, 20, 21, 22, and 23. The optimal classification threshold obtained by the NNA is 3.48. Under this configuration, the proposed RLMTS method correctly classifies all samples in the dataset.

Fig. 6 illustrates the MDs of normal and abnormal samples.



(a) MD of normal samples



(b) MD of abnormal samples

Fig.6. MDs of normal and abnormal samples under IR=1:1000.

Theoretically, RLMTS uses only a single class of samples as a baseline for constructing the MS and thus is largely independent of the imbalance ratio. RLMTS can achieve good diagnostic performance even if the IR is large.

4.5 Impact of data characteristics on RLMTS

Experiments are conducted to investigate the influence of data characteristics such as noise and feature correlation on the performance of RLMTS. In the noise analysis, SNR, standard deviation, and variance are used as evaluation metrics. For the feature correlation, distance correlation is used as the evaluation criterion.

(1) Data characteristics before and after RLMTS

Fig. 7 illustrates the original data characteristics of dataset C and the data characteristics after RLMTS processing.

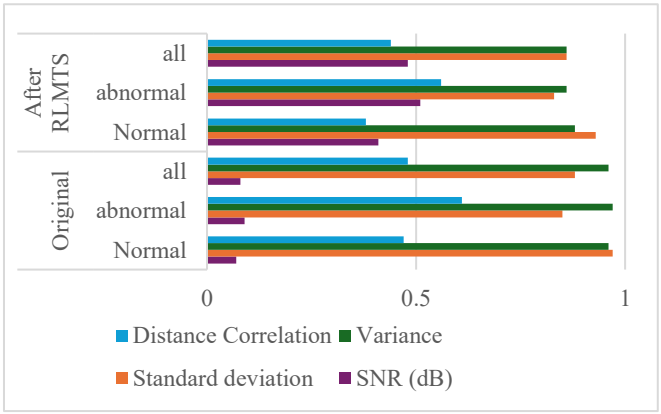


Fig.7. Data characteristics before and after RLMTS.

The average SNR values of the original data are 0.07 dB (normal class), 0.09 dB (abnormal class) and 0.08 dB (overall), which show a weak class discrimination and significant noise interference. After RLMTS processing, these SNR values significantly increase to 0.41 dB, 0.51 dB and 0.48 dB, indicating a clearer signal structure resulting from effective feature selection.

Meanwhile, both the standard deviation and variance have decreased, indicating that the model is more compact in feature expression after RLMTS processing, and

presents stronger anti-interference and stability. The decrease in standard deviation and variance also indirectly reflects an increase in the internal consistency of the samples.

To further analyze redundancy between the features, the distance correlation coefficient matrix is calculated between the features of each class of samples and the average of its upper triangular part is extracted as an evaluation metric. In the original data, the average distance correlation of the normal samples is 0.47, that of the abnormal samples is 0.61, and that of the whole is 0.48, indicating that there is a strong nonlinear dependency between the original features, which may affect the generalization ability of the model. After RLMTS processing, the distance correlation decreases to 0.39, 0.56, and 0.44, indicating that RLMTS effectively reduces the redundancy between the features and improves the discriminative capability, thereby improving the generalization capability of the model.

(2) Diagnostic performance of RLMTS on data after adding noise

To investigate the impact of data characteristics on RLMTS, particularly the effect of noise factors, Gaussian noise is added to the original data to construct a new Dataset C-4.

A lower magnitude of noise is added to the normal samples, while a larger magnitude is introduced to the abnormal samples. The increased perturbation in abnormal samples is intended to simulate the instability of real-world anomalies and potential sensor errors.

For the original sample X , let $\tilde{X}$ be the sample after adding noise.

$$\tilde{X} = \begin{cases} X + \phi_{normal}, & if\ X \in Normal\ class \\ X + \phi_{anomaly}, & if\ X \in Anomaly\ class \end{cases} \tag{20}$$

where $\phi_{normal} \sim N\left(0, (a_{normal} \cdot \sigma_{normal})^2\right)$ and $\phi_{anomaly} \sim N\left(0, (a_{anomaly} \cdot \sigma_{anomaly})^2\right)$, with σ_{normal} and $\sigma_{anomaly}$ denoting the standard deviations of normal and abnormal samples in each feature dimension, respectively. The noise scaling factors are empirically set as $a_{normal} = 0.03$ for normal samples and $a_{anomaly} = 0.08$ for abnormal samples.

The features with positive SNR gain, namely 3, 4, 6, 7, 9, 10, 11, 12, 14, 18, 20, 21, 22, 23, 24, 26, 29, 30, 31, 32, 33, and 36, constitute the initial feature space. After RL optimization, the selected feature subset becomes 2, 9, 14, 16, 20, 23, 24, 32, and 37. The NNA determines an optimal classification threshold of 3.74, yielding a confusion matrix of [7, 0; 0, 31]. These results indicate that RLMTS remains effective for fault diagnosis even in the presence of Gaussian noise.

4.6. Comparative models

To address class-imbalanced small-sample data, SMOTE is widely used to generate new samples. With advancements in deep learning, GAN has been employed to generate instances of underrepresented classes, significantly enhancing sample quality. SVM, logistic regression (LR), CNN, and recurrent neural networks (RNN) have also been widely used as effective classification models in various studies.

Therefore, 19 diagnostic models (15 widely used models and four state-of-the-art models) are selected for comparison with RLMTS. These models included SVM, LR, CNN, RNN, SMOTE-SVM, SMOTE-LR, SMOTE-CNN, SMOTE-RNN, GAN-SVM, GAN-LR, GAN-CNN, GAN-RNN, RL-CNN, RL-RNN and cost-sensitive LR. Additionally, DSADRSViT-IIRL [17] and ACWGAN-GP-LSTM [19] are included as state-of-the-art models. For further comparison, SMOTETomek-M1DCNN-SVM [36], which integrates SMOTETomek, multi-scale one-dimensional CNN, and SVM, is also considered for comparison. Ensemble method RUSBoost is also used for comparison.

The following subsections outline the hyperparameter optimization strategies employed to ensure fair model comparison.

(1) SMOTE

The G-value has a direct impact on the quality of the generated samples in SMOTE, where G represents the selection of G nearest neighbor samples. In our study, the G-value was tuned with a step of 1 in the interval [3,10] and the mean value of MD between the real and generated samples is used as a criterion. Fig. 8 shows the results, and the final choice is G = 8.

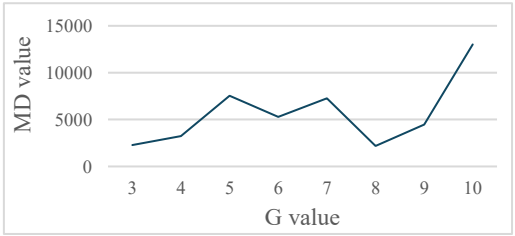


Fig. 8. MDs of samples with different G values from SMOTE.

(2) GAN

The latent space is essential in GANs as it captures the underlying structure of data, enabling the transformation of random noise into meaningful outputs. Therefore, the

dimension of latent space is tuned, which is explored in the range of 10-60 with a step size of 10. Fig. 9 shows the results by using the MD mean value between real and generated samples as a criterion. In this case, the final choice of latent dimension is 10.

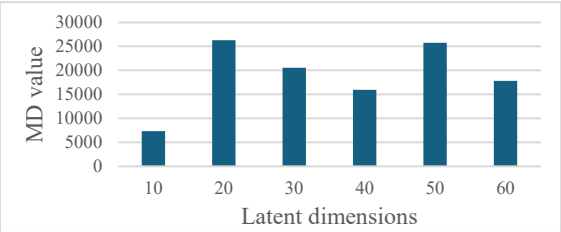


Fig. 9. MDs of samples with different latent dimensions from GAN.

(3) SVM

The hyperparameters such as BoxConstraint and KernelScale in the SVM are tuned. BoxConstraint is a constraint box, implying that the larger the value, the smaller the penalty, the smaller the interval of the final classification hyperplane, the more the support vectors, and the more complex the model. KernelScale is the geometric sequence of radial basis function sigma parameters scaled by the original kernel scale. These hyperparameters have a significant impact on the complexity and diagnostic accuracy of the SVM model. The process is shown in Fig. 10, with final BoxConstraint = 0.001 and KernelScale = 0.9867.

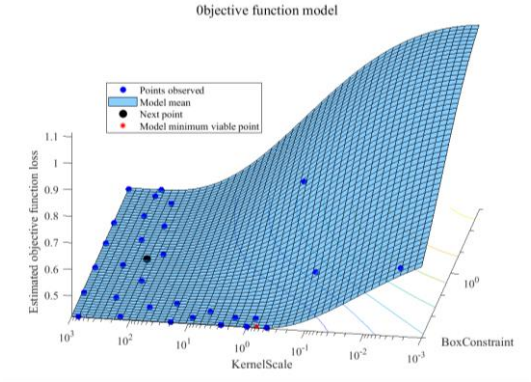


Fig. 10. SVM parameter optimization process.

(4) CNN

The hyperparameters, i.e., the kernel sizes of the convolutional layer and maximum pooling layer, as well as the dropout layer's discard probability in the CNN, are tuned. In our study, the kernel sizes of the convolutional layer and maximum pooling layer are set to 2 or 3, the discard probability of dropout layer is set to 0.1 or 0.2 or 0.3, and diagnosis accuracy is used as the evaluation criteria. Table 8 demonstrates the results. The final kernel size of the convolutional layer is 3, the kernel size of the maximum pooling layer is 2, and the dropout layer has a dropout probability of 0.2.

Table 8
Accuracy of CNN with different hyperparameters.

Convolutional layer kernel size	Max pooling layer kernel size	Dropout probability	Accuracy
2	2	0.1	0.59
		0.2	0.59
		0.3	0.54
	3	0.1	0.59
		0.2	0.57
		0.3	0.59
3	2	0.1	0.57
		0.2	0.61
		0.3	0.5902
	3	0.1	0.59
		0.2	0.59
		0.3	0.61

(5) RNN

The hidden layers of RNN are effective for feature extraction as the number of hidden layers not only affects the output at the current moment but also has an impact on future computations. In our study, the number of hidden layers in the RNN is tuned with a search space of 10-80 and a step size of 10, and the diagnostic accuracy of the training set is used as the evaluation criterion. Fig. 11 shows the results. The number of hidden layers is eventually chosen as 40.

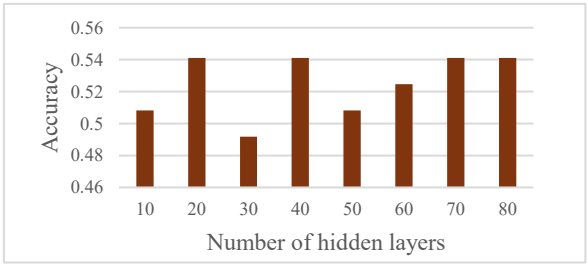


Fig. 11. Accuracy with different number of hidden layers for RNN.

4.7. Performance of the benchmark model with different small sample sizes

To evaluate the diagnostic robustness under limited data conditions, three new datasets named A-1, A-2, and A-3 are constructed by randomly selecting 20%, 40%, and 60% of the original data, respectively. The corresponding total sample sizes are 17, 35, and 52. For each dataset, 80% of the samples are used for training and the remaining 20% are used for validation. Nineteen benchmark models are then applied to these datasets to perform fault diagnosis. Table 9 presents the resulting confusion matrices, which

illustrate the diagnostic performance of the nineteen models on the three datasets

Table 9
Diagnostic performance of 19 benchmark models with different small samples.

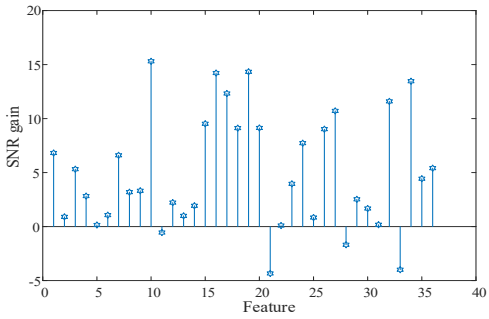
Dataset	Method	TP	FP	FN	TN	Time
A-1	RLMTS	1	0	0	2	4.66
	SVM	0	1	0	2	0.01
	LR	0	1	0	2	0.01
	CNN	0	1	0	2	0.02
	RNN	1	0	1	1	0.02
	SMOTE-SVM	0	1	0	2	0.01
	SMOTE-LR	0	1	0	2	0.02
	SMOTE-CNN	0	1	0	2	0.03
	SMOTE-RNN	1	0	1	1	0.05
	GAN-SVM	0	1	0	2	0.46
	GAN-LR	1	0	1	1	0.36
	GAN-CNN	0	1	0	2	0.34
	GAN-RNN	1	0	1	1	0.68
	RL-CNN	0	1	0	2	0.19
	RL-RNN	0	1	0	2	0.11
	Cost-sensitive LR	0	1	0	2	0.01
	DSADRSViT-IIRL	0	1	0	2	3.69
	ACWGAN-GP-LSTM	1	0	1	1	1.97
	SMOTETomek-M1DCNN-SVM	0	1	0	2	0.03
	RUSBoost	0	1	0	2	0.01
A-2	RLMTS	2	1	0	4	4.64
	SVM	0	3	0	4	0.01
	LR	0	3	0	4	0.01
	CNN	0	3	0	4	0.03
	RNN	0	3	0	4	0.04
	SMOTE-SVM	0	3	0	4	0.01
	SMOTE-LR	0	3	0	4	0.01
	SMOTE-CNN	0	3	0	4	0.04
	SMOTE-RNN	2	1	0	4	0.06
	GAN-SVM	0	3	0	4	0.26
	GAN-LR	0	3	0	4	0.24
	GAN-CNN	0	3	0	4	0.26
	GAN-RNN	2	1	0	4	0.51
	RL-CNN	0	3	0	4	0.23
	RL-RNN	1	2	1	3	0.10
	Cost-sensitive LR	0	3	1	3	0.01
	DSADRSViT-IIRL	1	2	0	4	6.03
	ACWGAN-GP-LSTM	2	1	3	1	2.84
	SMOTETomek-M1DCNN-SVM	1	2	2	2	0.05
	RUSBoost	1	2	0	4	0.01
A-3	RLMTS	4	0	0	6	4.76
	SVM	0	4	0	6	0.00
	LR	3	1	0	6	0.01
	CNN	0	4	0	6	0.04
	RNN	0	4	0	6	0.05
	SMOTE-SVM	1	3	0	6	0.01
	SMOTE-LR	3	1	0	6	0.01
	SMOTE-CNN	0	4	0	6	0.05
	SMOTE-RNN	0	4	0	6	0.08
	GAN-SVM	1	3	0	6	0.26
	GAN-LR	4	0	0	6	0.24
	GAN-CNN	0	4	0	6	0.27
	GAN-RNN	2	2	0	6	0.53
	RL-CNN	4	0	4	2	0.27
	RL-RNN	3	1	0	6	0.10
	Cost-sensitive LR	3	1	0	6	0.01
	DSADRSViT-IIRL	4	0	0	6	9.28
	ACWGAN-GP-LSTM	2	2	0	6	4.68
	SMOTETomek-M1DCNN-SVM	2	2	2	4	0.06
	RUSBoost	3	1	1	5	0.04

4.8. Case study

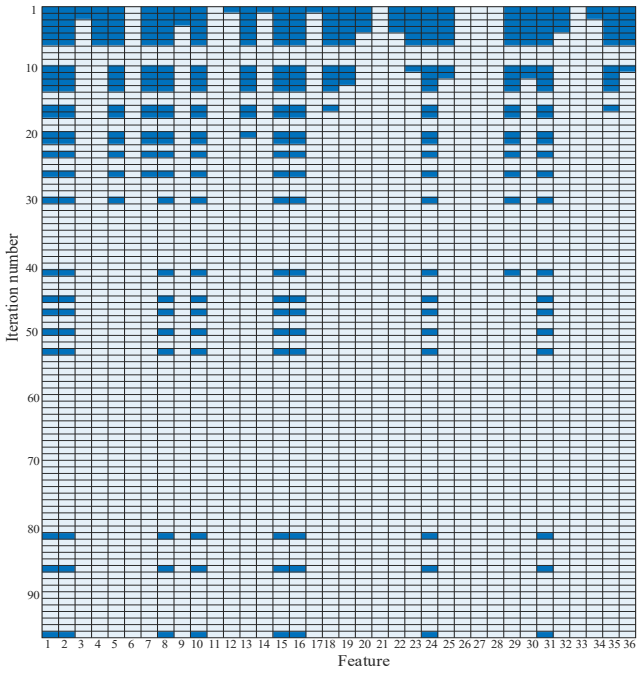
4.8.1. Case A with Dataset A

(1) MS construction

Fig. 12 presents the results of MS construction of Dataset A, with (a) illustrating the SNR gain obtained using orthogonal array, and (b) showing the action change process using RL and the added action execution rules (set K=1 as the example). Note that in Fig. 12 (a), the SNR gains between the variables are independent of each other, and their values and the shape of plot depend on the variable values. Blue color in Fig. 12 (b) indicates the selected variables while white color indicates those unselected variables.



(a) SNR gain



(b) Agent action change trajectory

Fig. 12. Results of MS construction.

(2) Threshold determination

The optimal thresholds are calculated using NNA based on the selected parameters by the above MS. Set $\xi = 1$ as an example, the trajectories of the optimal thresholds are shown in Fig. 13.

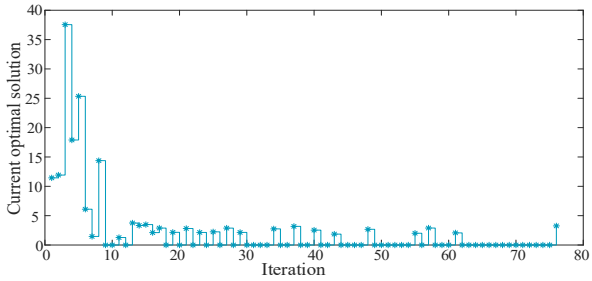


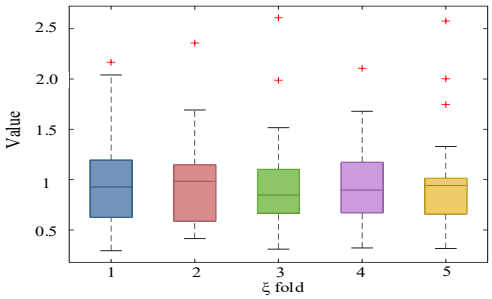
Fig. 13. Variation of the current optimal solution.

In each iteration, a current optimal solution is obtained and compared with the historical best. The global optimal solution is updated accordingly based on the minimization of the fitness function.

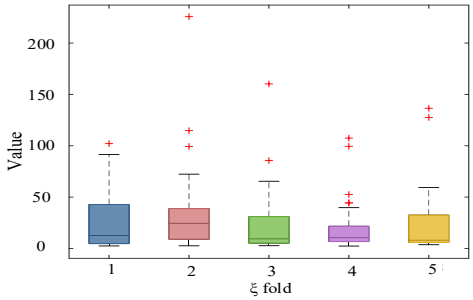
The final global optimal thresholds obtained from five-fold cross-validation on Dataset A are 2.91, 2.89, 2.65, 2.26, and 2.93, respectively.

(3) Classification results of training samples

Fig. 14 presents the boxplots of normal and abnormal samples in the training set under five-fold cross-validation.



(a) MDs of normal samples



(b) MDs of abnormal samples

Fig. 14. boxplots of normal and abnormal samples on Dataset A.

As shown in Fig. 14, there is a clear distinction in MDs between normal and abnormal samples, which facilitates the subsequent classification task performed by the RLMTS model.

(4) Classification results of test samples

The MDs of the testing samples are calculated based on the MS. These MDs are then used to compare with the threshold, as described in Section 3.4, thereby generating the classification results.

Table 10 presents the diagnostic performance of RLMTS on both the training and testing sets of Dataset A. In the training set, all samples are correctly classified except for two faulty samples misclassified as normal in the first fold and one in the second fold. On the testing set, RLMTS achieves high diagnostic accuracy for all samples when $\xi = 1$, 2, and 3. The relatively low recall observed at $\xi = 4$ is primarily due to the small number of total samples, despite the RLMTS misclassifying only two. Similarly, the low precision at $\xi = 5$ is due to the same reason, as only one sample is misclassified. Overall, RLMTS exhibits strong diagnostic capability.

Table 10

Diagnostic performance of the RLMTS on Dataset A.

Data	ξ	precision	recall	specificity	accuracy	F1	G-mean
Training set	1	0.94	1.00	0.95	0.97	0.97	0.98
	2	0.96	1.00	0.97	0.98	0.98	0.99
	3	1.00	1.00	1.00	1.00	1.00	1.00
	4	1.00	1.00	1.00	1.00	1.00	1.00
	5	1.00	1.00	1.00	1.00	1.00	1.00
	Mean	0.98	1.00	0.98	0.99	0.99	0.99
Testing set	1	1.00	1.00	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	1.00	1.00	1.00
	3	1.00	1.00	1.00	1.00	1.00	1.00
	4	1.00	0.71	1.00	0.87	0.83	0.85
	5	0.88	1.00	0.90	0.94	0.93	0.95
	Mean	0.98	0.94	0.98	0.96	0.95	0.96

(5) Comparison with benchmark methods

The average value of the five-fold cross-validation is used for comparison. Table 11 illustrates the comparison results of the proposed method with 19 benchmark models. As shown, RLMTS evidently exhibits higher precision, recall, accuracy, F1, and G-mean values, demonstrating its superior diagnostic performance to these benchmark models. A comprehensive analysis of the model's performance and its contributing factors is provided in Section 4.9.

4.8.2. Case B with Dataset B

(1) MS construction and threshold determination

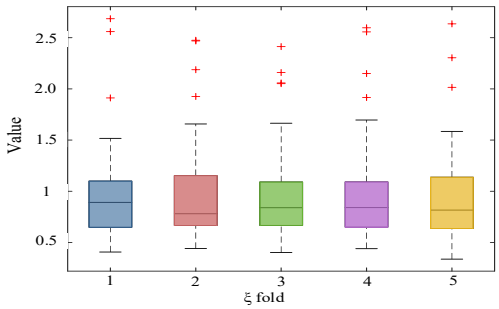
After being filtered with SNR gain, 14 variables (Number 1, 2, 3, 4, 5, 10, 15, 18, 19, 21, 25, 27, 30 and 37) with SNR gain less than 0 are eliminated.

After the RL process, the selected feature indices are 4, 11, 14, 15, 16, 20, 22, 24, 26, 28, 30, 34, and 35.

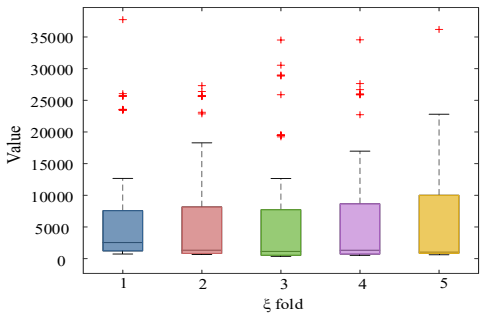
The final global optimal thresholds obtained from five-fold cross-validation on Dataset B are 443.71, 478.37, 130.19, 506.95 and 586.20, respectively.

(2) Classification results of training samples

Fig. 15 presents the boxplots of normal and abnormal samples in the training set under five-fold cross-validation.



(a) MDs of normal samples



(b) MDs of abnormal samples

Fig. 15. boxplots of normal and abnormal samples on Dataset B.

As shown in Fig. 15, in the five-fold cross-validation on the training set, the mean MD values for normal samples in the testing subsets calculated by the RLMTS model are 0.98, 0.98, 0.98, 0.98, and 0.97, respectively, while the corresponding values for abnormal samples are 6511.83, 5869.62, 5308.31, 5674.89, and 5279.03. The substantial difference in MD between normal and abnormal samples facilitates the accurate sample discrimination.

(3) Classification results of test samples

Table 12 presents the diagnostic performance of the RLMTS on both the training and testing sets of Dataset B.

As shown in Table 12, the RLMTS achieves high accuracy diagnostic performance for all samples. The high diagnostic accuracy of the proposed method is explained in the Discussion section.

Table 12

Diagnostic performance of the RLMTS on Dataset B.

Data	ξ	precision	recall	specificity	accuracy	F1	G-mean
Training set	1	1.00	1.00	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	1.00	1.00	1.00
	3	1.00	1.00	1.00	1.00	1.00	1.00
	4	1.00	1.00	1.00	1.00	1.00	1.00
	5	1.00	1.00	1.00	1.00	1.00	1.00
	Mean	1.00	1.00	1.00	1.00	1.00	1.00
Testing set	1	1.00	1.00	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	1.00	1.00	1.00
	3	1.00	1.00	1.00	1.00	1.00	1.00
	4	1.00	1.00	1.00	1.00	1.00	1.00
	5	1.00	1.00	1.00	1.00	1.00	1.00
	Mean	1.00	1.00	1.00	1.00	1.00	1.00

(4) Comparison with benchmark methods

Table 13 shows the comparison results of the proposed method with 19 benchmark models. It can be seen that RLMTS has a highly accurate diagnostic performance compared to these competitor models, demonstrating that the RLMTS is more suitable for flow meter diagnosis with class-imbalanced small-sample data. A comprehensive analysis of the model's performance and its contributing factors is provided in Section 4.9.

4.8.3. Case C with Dataset C

(1) MS construction and threshold determination

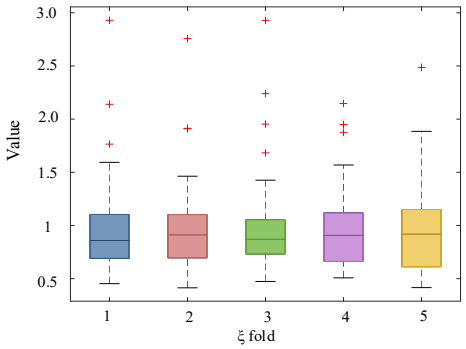
After SNR gain selection, 18 variables (number 1, 2, 3, 4, 5, 10, 15, 18, 20, 25, 26, 27, 29, 30, 31, 33, 34 and 37) with SNR gain less than 0 are removed.

After the RL process, the selected feature indices are 6, 7, 8, 11, 13, 14, 16, 19, 23, 35, and 37.

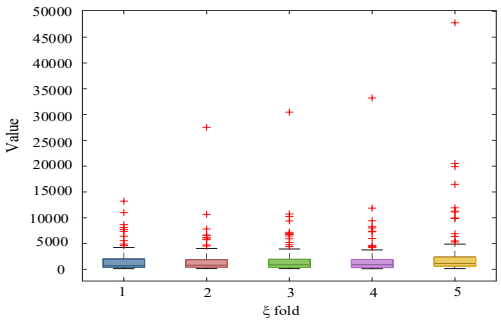
Based on five-fold cross-validation using NNA, the final optimal thresholds used for classifying the testing samples are 70.19, 65.39, 67.79, 66.76, and 83.36, respectively.

(2) Classification results of training samples

Fig. 16 presents the boxplots of normal and abnormal samples in the training set under five-fold cross-validation.



(a) MDs of normal samples



(b) MDs of abnormal samples

Fig. 16. boxplots of normal and abnormal samples on Dataset C.

As shown in Fig. 16, in the five-fold cross-validation on training set, the mean values of MD for normal samples calculated by the RLMTS model are 0.98, 0.98, 0.98, 0.97, 0.97, while the mean values of MD for abnormal samples are 1791.28, 1805.08, 1921.23, 1915.98, and 2897.14, respectively. The difference in MDs between normal and abnormal samples appears very significant.

(3) Classification results of test samples

Table 14 presents the diagnostic performance of the RLMTS on both the training and testing sets of Dataset C. As shown in Table 14, the RLMTS achieves correct classification of all samples in each fold of the cross-validation. The high diagnostic accuracy of the proposed method is explained in the Discussion section.

Table 14

Diagnostic performance of the RLMTS on Dataset C.

Data	ξ	precision	recall	specificity	accuracy	F1	G-mean
Training set	1	1.00	1.00	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	1.00	1.00	1.00
	3	1.00	1.00	1.00	1.00	1.00	1.00
	4	1.00	1.00	1.00	1.00	1.00	1.00
	5	1.00	1.00	1.00	1.00	1.00	1.00
	Mean	1.00	1.00	1.00	1.00	1.00	1.00
Testing set	1	1.00	1.00	1.00	1.00	1.00	1.00
	2	1.00	1.00	1.00	1.00	1.00	1.00
	3	1.00	1.00	1.00	1.00	1.00	1.00
	4	1.00	1.00	1.00	1.00	1.00	1.00
	5	1.00	1.00	1.00	1.00	1.00	1.00
	Mean	1.00	1.00	1.00	1.00	1.00	1.00

(4) Comparison with benchmark methods

Table 15 shows the comparison results of the proposed method with 19 benchmark models. As shown, RLMTS presents a highly accurate diagnostic performance compared to other models on Dataset C.

The reason for the high accuracy of RLMTS on Dataset C lies in the significant difference in MD between the two classes. The RNN-related methods perform poorly on Dataset C, judging all samples as abnormal samples. This is possibly because the sample size of the abnormal samples is larger than that of the normal samples, as well as the quality of the generated samples. More details can be found in Section 4.9.

Table 11

Diagnostic performance of different methods on Dataset A.

Method	precision	recall	specificity	accuracy	F1	G-mean	Time (second)
RLMTS	0.98	0.94	0.98	0.96	0.95	0.96	4.73
SVM	0.00	0.00	1.00	0.59	0.00	0.00	0.01
LR	0.00	0.00	1.00	0.59	0.00	0.00	0.01
CNN	0.60	0.67	0.57	0.62	0.53	0.54	0.07
RNN	0.09	0.06	0.93	0.57	0.07	0.09	0.03
SMOTE-SVM	0.70	0.40	0.86	0.68	0.44	0.46	0.04
SMOTE-LR	0.46	0.48	0.59	0.54	0.46	0.53	0.04
SMOTE-CNN	0.46	0.55	0.49	0.52	0.47	0.48	0.12
SMOTE-RNN	0.22	0.14	0.72	0.49	0.16	0.22	0.06
GAN-SVM	0.60	0.24	1.00	0.69	0.31	0.36	1.21
GAN-LR	0.70	0.15	0.95	0.61	0.24	0.38	1.21
GAN-CNN	0.44	0.67	0.39	0.50	0.47	0.39	1.28
GAN-RNN	0.31	0.37	0.61	0.50	0.32	0.38	1.24
RL-CNN	0.41	0.58	0.55	0.57	0.45	0.34	0.46
RL-RNN	0.30	0.56	0.36	0.47	0.38	0.16	0.30
Cost-sensitive LR	0.41	1.00	0.00	0.41	0.58	0.00	0.01
DSADRSViT-IIRL	0.77	0.54	0.87	0.73	0.62	0.68	6.53
ACWGAN-GP-LSTM	0.60	0.90	0.16	0.58	0.71	0.27	2.80

Method	precision	recall	specificity	accuracy	F1	G-mean	Time (second)
SMOTETomek-M1DCNN-SVM	0.45	0.28	0.64	0.49	0.30	0.40	0.09
RUSBoost	0.86	0.77	0.90	0.85	0.81	0.83	0.04

Table 13

Diagnostic performance of different methods on Dataset B.

Method	Precision	Recall	Specificity	Accuracy	F1	G-mean	Time (second)
RLMTS	1.00	1.00	1.00	1.00	1.00	1.00	4.87
SVM	0.75	0.89	0.88	0.88	0.80	0.88	0.01
LR	0.75	0.72	0.88	0.83	0.72	0.79	0.01
CNN	0.30	1.00	0.00	0.30	0.45	0.00	0.12
RNN	0.00	0.00	0.99	0.69	0.00	0.00	0.06
SMOTE-SVM	0.49	1.00	0.56	0.69	0.65	0.74	0.03
SMOTE-LR	0.00	0.00	1.00	0.70	0.00	0.00	0.03
SMOTE-CNN	0.89	0.98	0.95	0.96	0.93	0.96	0.19
SMOTE-RNN	0.87	0.48	0.96	0.82	0.60	0.67	0.10
GAN-SVM	0.52	1.00	0.62	0.73	0.68	0.78	5.62
GAN-LR	0.58	0.28	0.95	0.75	0.37	0.45	5.62
GAN-CNN	0.65	0.11	0.96	0.71	0.18	0.29	5.77
GAN-RNN	0.63	0.47	0.85	0.75	0.53	0.63	5.69
RL-CNN	0.30	1.00	0.00	0.30	0.45	0.00	0.66
RL-RNN	0.30	1.00	0.00	0.30	0.45	0.00	0.36
Cost-sensitive LR	0.30	1.00	0.00	0.30	0.45	0.00	0.01
DSADRSViT-IIRL	0.89	0.97	0.94	0.95	0.93	0.96	13.05
ACWGAN-GP-LSTM	0.97	1.00	0.97	0.98	0.98	0.98	1.73
SMOTETomek-M1DCNN-SVM	0.55	0.98	0.66	0.76	0.70	0.80	0.22
RUSBoost	0.94	0.99	0.97	0.98	0.96	0.98	0.01

Table 15

Diagnostic performance of different methods on Dataset C.

Method	Precision	Recall	Specificity	Accuracy	F1	G-mean	Time (second)
RLMTS	1.00	1.00	1.00	1.00	1.00	1.00	4.77
SVM	0.82	0.88	0.92	0.90	0.84	0.90	0.01
LR	0.75	0.68	0.90	0.83	0.71	0.78	0.01
CNN	0.28	1.00	0.00	0.28	0.44	0.00	0.12
RNN	0.00	0.00	1.00	0.72	0.00	0.00	0.06
SMOTE-SVM	0.38	1.00	0.35	0.54	0.54	0.58	0.02
SMOTE-LR	0.00	0.00	1.00	0.72	0.00	0.00	0.02
SMOTE-CNN	0.95	0.66	0.99	0.89	0.75	0.79	0.18
SMOTE-RNN	0.76	0.46	0.93	0.79	0.56	0.64	0.09
GAN-SVM	0.55	1.00	0.66	0.76	0.70	0.81	6.04
GAN-LR	0.69	0.29	0.95	0.76	0.40	0.51	6.05
GAN-CNN	0.10	0.03	0.99	0.71	0.04	0.07	6.20
GAN-RNN	0.84	0.41	0.97	0.81	0.54	0.62	6.11
RL-CNN	0.28	1.00	0.00	0.28	0.44	0.00	0.71
RL-RNN	0.28	1.00	0.00	0.28	0.44	0.00	0.35
Cost-sensitive LR	0.28	1.00	0.00	0.28	0.44	0.00	0.01
DSADRSViT-IIRL	0.94	0.99	0.98	0.98	0.96	0.98	11.69
ACWGAN-GP-LSTM	0.97	1.00	0.97	0.98	0.98	0.98	2.45

Method	Precision	Recall	Specificity	Accuracy	F1	G-mean	Time (second)
SMOTETomek-M1DCNN-SVM	0.88	1.00	0.90	0.94	0.93	0.95	0.21
RUSBoost	0.95	1.00	0.99	0.99	0.97	0.99	0.01

4.9. Discussions

The ablation study in Section 4.3 demonstrates the effectiveness of key components in RLMTS, including RL, NNA, SNR gain, and ε-greedy strategy. These modules are indispensable to the framework, and their synergistic integration provides RLMTS with not only enhanced diagnostic performance but also strong robustness.

The experiments in Section 4.4 verify that RLMTS is well suited for fault diagnosis tasks under varying imbalance ratios and limited sample conditions. Section 4.5 further confirms that RLMTS is capable of effectively handling datasets with different data characteristics. RLMTS, therefore, exhibits broad applicability and strong generalization capability.

The comparative results in Sections 4.7 and 4.8 show that RLMTS outperforms 19 benchmark models and is particularly well suited for imbalanced and small sample fault diagnosis tasks.

The reasons for high performance of RLMTS are as follows: (i) RLMTS constructs the MS based on one class of samples, which is less affected by the imbalance ratio. (ii) RLMTS optimizes the MS using RL. Specifically, the feature space selected based on SNR gain is used as the initial action space, which helps mitigate the influence of noise. Through continuous interaction with the environment, rewards are calculated, and the feature space is optimized according to the reward maximization principle (i.e., minimizing the objective function defined in eq (8)) and a given exploration strategy. This optimization process effectively enlarges the MD difference between normal and abnormal samples, which is beneficial for downstream classification. (iii) RLMTS introduces the NNA to determine the optimal threshold. After initializing the population and calculating the initial fitness values, NNA simulates neural connection propagation through a weight matrix to generate new solutions, evaluates their fitness, and introduces perturbations during the process. In each iteration, the search range is gradually reduced to enhance convergence, and the objective function is updated while preserving the historically best solution. By supporting global search and avoiding local optima, RLMTS can adaptively determine the optimal threshold within a predefined search range to achieve effective sample classification.

Compared with the benchmark methods, RLMTS has an average computation time of approximately 4.80 seconds, which is higher than that of some baseline algorithms but is not the highest overall. Given its great diagnostic accuracy, the computation time of 4.80 seconds is acceptable and well justified. Note that the computation time reported in the performance tables is measured using programs executed in Python 3.10.12 on a system equipped with a 12th Gen Intel® Core™ i7-12700H processor (up to 3.69 GHz), 16 GB of DDR5 RAM (4800 MT/s), and an NVIDIA GeForce RTX 3060 Laptop GPU (6 GB VRAM).

In Table 11, the specificity value of RLMTS (0.98) is slightly lower than that of SVM, LR, and GAN-SVM (1.00). Although these benchmark models exhibit great discrimination against negative class samples, their ability to correctly identify positive class samples is significantly inferior to that of RLMTS.

In Dataset A, both SVM and LR misclassify all test samples as negative class due to insufficient training data and class imbalance, which collectively prevent effective model training. Although RNN outperforms SVM and LR in diagnostic capability, it still demonstrates suboptimal performance with a persistent negative classification bias. This suggests that while the network architecture could extract meaningful features for classification tasks, the combined challenges of small sample size and class imbalance significantly constrain its training efficacy. CNN performs relatively better than RNN in diagnosis but remains unsatisfactory overall, primarily due to its reliance on large-scale training data. After applying SMOTE and GAN techniques to address class imbalance, notable improvements are observed: (i) balanced class distribution and increased training sample quantity can enhance SVM and LR performance, and (ii) RNN achieves higher positive class diagnosis rates. However, the generated samples introduce additional noise, resulting in slight performance degradation for CNN. ACWGAN-GP-LSTM exhibits a distinct positive classification bias despite maintaining some negative class discernment. This skewed preference likely originates from deep networks' heightened sensitivity to training data quality and quantity requirements, ultimately constraining overall accuracy. The inferior performance of SMOTETomek-M1DCNN-

SVM stems from two factors: (i) the noise introduced during SMOTETomek sample generation impairs the model effectiveness; (ii) the inadequate training sample size limits the M1DCNN's feature extraction capability.

In Datasets B and C, SVM and LR have higher diagnostic performance, CNN tends to diagnose the test sample as “positive class”, while RNN tends to output “negative class”. This divergence likely originates from the inherent differences in feature extraction capabilities among network architectures when handling data-constrained scenarios with class imbalance. Notably, both CNN and RNN demonstrate suboptimal diagnostic efficacy under these conditions. When applying SMOTE and GAN to deal with class-imbalanced small-sample data, SVM exhibits a tendency to predict "positive class" while LR leans toward "negative class" outcomes, suggesting that the noise introduced during pseudosample generation negatively impacts models with originally robust diagnostic capabilities, thereby counteracting their performance. In contrast, the generated samples effectively address training data scarcity and class imbalance issues for CNN and RNN, leading to an improved diagnostic performance. Remarkably, SMOTE demonstrates greater effectiveness than GAN in enhancing these deep learning models, which may be attributed to its simpler implementation with fewer hyperparameters compared to GAN's complex architecture which involves multiple parameters and network training requirements, a configuration particularly challenging to optimize under limited sample conditions.

Among the SVM-related, LR-related, CNN-related (except RL-CNN) and RNN-related (except RL-RNN) methods, the SVM-related methods have the best overall diagnostic performance for class-imbalanced small-sample data (average accuracy of 0.72, average F1 of 0.55 and average G-mean of 0.61). Among the remaining three categories, LR-related methods present the highest accuracy (0.71) while CNN-related (except RL-CNN) methods have the highest F1 (0.47) and G-mean values (0.39).

The diagnostic performance of RL-CNN and RL-RNN is not improved after optimizing CNN and RNN using RL due to the fact that (i) both CNN and RNN need a larger sample size to learn effective features, and the small sample size in this study cannot satisfy the requirement of effective learning; and (ii) the optimization strategy space of RL is inherently complex when it is used for network parameter optimization. Under conditions of limited samples and class imbalance, the models are still prone to overfitting the majority class or encountering unstable training, even when higher weights are assigned to the minority class during training.

After using SMOTE to generate samples, CNN and RNN have improved their diagnostic ability for minority class samples, and have increased the accuracy, F1 and G-mean values by 0.39, 0.24 and 0.56 for CNN, 0.04, 0.42 and 0.48 for RNN. This indicates that the use of SMOTE to generate samples is effective for classification with neural networks under class-imbalanced small-sample data.

After using GAN to generate pseudo-samples, SVM-related and CNN-related diagnostic ability for minority class samples has been improved whereas LR-related performance has instead been decreased. RNN-related (except RL-RNN) comprehensive indicator values have all increased (accuracy increased by 0.02, F1 value increased by 0.44, and G-mean value increased by 0.51). This suggests that the samples generated by the GAN are not always consistent for these classifiers.

Although cost-sensitive LR incorporates class weights, it predicts all samples as belonging to a single class. This may be attributed to the limited discriminative power of the features, if the feature distributions between classes are not sufficiently distinct. LR, as a linear classifier, may fail to learn an effective decision boundary and degenerate into predicting only one class.

RUSBoost achieves performance metrics above 0.9 on Datasets B and C, indicating strong diagnostic capability; however, its poor performance on Dataset A suggests that the stability and generalization ability of RUSBoost still require further improvement.

Compared to the three state-of-the-art fault diagnosis models (DSADRSViT-IIRL, ACWGAN-GP-LSTM and SMOTETomek-M1DCNN-SVM), the RLMTS is more suitable for flow meter fault diagnosis of class-imbalanced small-sample data, and it has better robustness and stability. The reasons for the better diagnostic performance of the RLMTS are as follows: (i) RLMTS does not rely on expanding the number of samples for small samples and class imbalance problems, so it does not introduce new noise in the diagnosis process; (ii) RLMTS fully leverages the exploration capabilities of RL, effectively removing redundant information while retaining important data; (iii) RLMTS uses a single-class sample as a baseline for fault diagnosis, which is less affected by the class imbalance of the data.

In summary, when addressing small sample and class imbalance problems, existing mainstream methods

generally face the following core challenges: (i) Decision boundary degradation: linear models such as LR tend to collapse into predicting a single class when the features are not linearly separable. (ii) Sensitivity to data augmentation: techniques like SMOTE and GAN may introduce noise during the generation of synthetic samples, which can negatively affect generalization performance. (iii) Dependence on data scale: models such as CNN and RNN require large amounts of data to effectively extract representative features. (iv) Unstable generalization performance: for example, RUSBoost exhibits significant performance fluctuations across different datasets.

In contrast, RLMTS does not rely on data augmentation, exhibits stable feature selection, converges quickly during training, and demonstrates strong discriminative capability for both positive and negative samples. It consistently achieves superior performance across multiple datasets, striking a well-balanced trade-off among model complexity, robustness, and diagnostic accuracy.

5. Conclusion

The RLMTS model is proposed to address the class-imbalanced small-sample data of flow meter fault diagnosis. In the MS construction module, RLMTS takes the feature subset composed of variables with SNR gain greater than zero as the initial MS and constructs the effective MS through the exploration of agents from RL. During MS optimization, at each iteration, the agent generates new actions based on the previous iteration's results, incorporating the ε greedy algorithm and predefined rules. The agent aims to maximize rewards (i.e., minimizing eq. (8)) to construct an effective MS. In the threshold determination and fault diagnosis modules, RLMTS searches for the optimal thresholds using NNA. The NNA explores the solution space via swarm intelligence, computing fitness function values for the current solutions. Through iterative optimization, it identifies the optimal solution along with its corresponding fitness value, thereby determining the final optimal threshold. The classes of test samples are determined based on the optimal threshold and their MDs.

To verify the effectiveness of RLMTS, three practical datasets with different IRs are selected for experiments, which are cross-validated using a five-fold cross validation approach.

The IRs of the three datasets are 1:1.49, 1:2.35, and 1:2.53, respectively, and 20%, 40%, and 60% of their data are taken for the experiment. The accuracy values of RLMTS are 0.95, 1.0, and 0.98, F1 values are 0.93, 1.0, and 0.96, and G-mean values are 0.94, 1.0 and 0.98, indicating that RLMTS is suitable for flow meter fault diagnosis with different IRs.

The mean values of precision, recall, specificity, accuracy, F1, and G-mean for 19 competitor models on three datasets are analyzed. The performance indicator values of RLMTS are higher than those of 19 benchmark models, demonstrating that RLMTS has a better diagnostic performance and is more robust and thus applicable to a wider range of applications.

The comparative analysis with the benchmark models demonstrates the theoretical advancement that the RLMTS has made. RLMTS effectively combines MTS, RL and NNA to make full use of their respective advantages, providing a new research method for flow meter fault diagnosis under class-imbalanced small samples. Different from existing methods, RLMTS does not expand the training set by generating new samples, but uses the original data to ensure that no new noise is introduced during data processing. Meanwhile, the difference between MDs of normal and abnormal samples is maximized by fully mining the features to finally achieve fault diagnosis.

RLMTS extends the application of the MTS to small sample imbalanced scenarios and overcomes the limitation of the diagnostic model for data size, being able to effectively deal with flowmeter fault diagnosis, and also provides a new framework for the fault diagnosis of other devices.

From the practical perspective, RLMTS provides a new diagnostic framework and model for class-imbalanced small-sample data with a wide range of applications. More practical datasets under different operational scenarios would be required to further ensure the stability, reliability and robustness of the proposed approach. In addition, future work will extend RLMTS to accommodate more complex data types, such as image-based information and high-frequency vibration signals, thereby enabling its application to a broader range of industrial fault diagnosis scenarios.

Declaration of competing interest

No potential conflict of interest was reported by the author(s).

Acknowledgments

This work was supported by China Scholarship Council (202406840086) and National Natural Science Foundation of China (No. 72401080). The authors would also like to thank the support from Royal Society International Exchanges under Grant IEC\NSFC\223091.

Data Availability

The authors do not have permission to share data.

References

[1] P. Mohindru, Recent advancements in volumetric flow meter for industrial application, Heat Mass Transf. 59 (2023) 2149–2166. https://doi.org/10.1007/s00231-023-03413-4.

[2] H. Cai, W. Liu, K. Zhou, X. Wang, K. Lin, X.-Y. Tang, Physics constrained high-precision data-driven modeling for multi-path ultrasonic flow meter in natural gas measurement, Sensors 24 (2024). https://doi.org/10.3390/s24144521.

[3] A.N. Johnson, E. Harman, J.T. Boyd, Blow-down calibration of a large ultrasonic flow meter, Flow Meas. Instrum. 77 (2021). https://doi.org/10.1016/j.flowmeasinst.2020.101848.

[4] Z. Chen, W. Zhao, P. Shen, C. Wang, Y. Jiang, A Fault diagnosis method for ultrasonic flow meters based on KPCA-CLSSA-SVM, Processes 12 (2024). https://doi.org/10.3390/pr12040809.

[5] H. Oh, Y.-Y. Choi, M. Kim, Y.-J. Sohn, S.-G. Kim, W.-Y. Lee, Experimental validation of passive and active fault-tolerant controls against sensor faults in a proton exchange membrane fuel cell system, J Process Control 129 (2023). https://doi.org/10.1016/j.jprocont.2023.103064.

[6] R. Liu, Y. Zhang, Z. Li, Leakage diagnosis of air conditioning water system networks based on an improved BP neural network algorithm, Buildings 12 (2022). https://doi.org/10.3390/buildings12050610.

[7] S. Qiu, Z. Li, R. He, Z. Li, User-friendly fault detection method for building chilled water flowmeters: Field data validation, Sci Technol Built Environ 28 (2022) 1116–1137. https://doi.org/10.1080/23744731.2022.2043068.

[8] H. Pan, B. Li, J. Zheng, J. Tong, Q. Liu, S. Deng, Research on roller bearing fault diagnosis based on robust smooth constrained matrix machine under imbalanced data, Adv. Eng. Inform. 62 (2024). https://doi.org/10.1016/j.aei.2024.102667.

[9] F. Thabtah, S. Hammoud, F. Kamalov, A. Gonsalves, Data imbalance in classification: Experimental evaluation, Inf Sci (N Y) 513 (2020) 429–441. https://doi.org/10.1016/j.ins.2019.11.004.

[10] Y.-L. He, Q. Hua, Q.-X. Zhu, S. Lu, Enhanced virtual sample generation based on manifold features: Applications to developing soft sensor using small data, ISA Trans 126 (2022) 398–406. https://doi.org/10.1016/j.isatra.2021.07.033.

[11] E. Kim, S. Jung, M. Kim, J. Kim, B. Kim, J. Kim, S. Kim, Anomaly detection using puzzle-based data augmentation to overcome data imbalances and deficiencies, Machines 11 (2023). https://doi.org/10.3390/machines11111034.

[12] J. Lu, W. Wu, X. Huang, Q. Yin, S. Yang Kuangzhi and Li, A modified active learning intelligent fault diagnosis method for rolling bearings with unbalanced samples, Adv. Eng. Inform. 60 (2024). https://doi.org/10.1016/j.aei.2024.102397.

[13] A. Fazli, J. Poshtan, Wind turbine fault prognosis using SCADA measurements, pre-fault labeling, and KNN classifiers robust against data imbalance, Meas. 243 (2025). https://doi.org/10.1016/j.measurement.2024.116202.

[14] J.Y. Chen, Z.T. Yan, C.Y. Lin, B.Q. Yao, H.J. Ge, Aero-engine high speed bearing fault diagnosis for data imbalance: a sample enhanced diagnostic method based on pre-training WGAN-GP, Meas. 213 (2023). https://doi.org/10.1016/j.measurement.2023.112709.

[15] N. Zhang, L. Duan, X. Fan, Fault diagnosis of planetary gearboxes under small and imbalanced samples based on enhanced Siamese network with improved downsampling module, J. Vib. Control (2024). https://doi.org/10.1177/10775463241288058.

[16] H. Fan, J. Ma, X. Cao, X. Zhang, Q. Mao, An intelligent diagnosis approach combining resampling and CWGAN-GP of single-to-mixed faults of rolling bearings under unbalanced small samples, Intern J Pattern Recognit Artif Intell 37 (2023). https://doi.org/10.1142/S0218001423560177.

[17] L. Zhang, S. Gu, H. Luo, L. Ding, Y. Guo, Residual shrinkage ViT with discriminative rebalancing strategy for small and imbalanced fault diagnosis, Sensors 24 (2024). https://doi.org/10.3390/s24030890.

[18] Y.B. Li, W.T. Zou, L. Jiang, Fault diagnosis of rotating machinery based on combination of

Wasserstein generative adversarial networks and long short term memory fully convolutional network, Meas. 191 (2022). https://doi.org/10.1016/j.measurement.2022.110826.

[19] K. Shen, D. Zhao, A Fault diagnosis method under data imbalance based on generative adversarial network and long short-term memory algorithms for aircraft hydraulic system, Aerospace 10 (2023). https://doi.org/10.3390/aerospace10020164.

[20] J. Wang, J. Wei, H. Huang, L. Wen, Y. Yuan, H. Chen, R. Wu, J. Wu, IMWMOTE: A novel oversampling technique for fault diagnosis in heterogeneous imbalanced data, Expert Syst Appl 251 (2024). https://doi.org/10.1016/j.eswa.2024.123987.

[21] F. Li, Y. Min, Y. Zhang, Y. Zhang, H. Zuo, F. Bai, Evaluation method for consistency of lithium-ion battery packs in electric vehicles based on the Mahalanobis-Taguchi system, J Energy Storage 78 (2024). https://doi.org/10.1016/j.est.2023.110045.

[22] L.M. Tan, W.Z.A. Wan Muhamad, A.K. Yahya Zainor Ridzuan and Junoh, N.H.A. Azziz, F. Ramlie, N. Harudin, M.Y. Abu, X.J. Tan, A survey on improvement of Mahalanobis Taguchi system and its application, Multimed Tools Appl 82 (2023) 43865–43881. https://doi.org/10.1007/s11042-023-15257-5.

[23] F. Ramlie, W.Z.A.W. Muhamad, N. Harudin, M.Y. Abu, H. Yahaya, K.R. Jamaludin, H.H. Abdul Talib, Classification performance of thresholding methods in the Mahalanobis-Taguchi System, Appl. Sci. 11 (2021). https://doi.org/10.3390/app11093906.

[24] S. Fan, X. Zhang, Z. Song, Imbalanced sample selection with deep reinforcement learning for fault diagnosis, IEEE Trans Industr Inform 18 (2022) 2518–2527. https://doi.org/10.1109/TII.2021.3100284.

[25] Y. Li, Y. Wang, X. Zhao, Z. Chen, A deep reinforcement learning-based intelligent fault diagnosis framework for rolling bearings under imbalanced datasets, Control Eng Pract 145 (2024). https://doi.org/10.1016/j.conengprac.2024.105845.

[26] Z. Zhu, W. Wu, T. Chen, J. Hu, C. Wang, Integrating reinforcement learning with deterministic learning for fault diagnosis of nonlinear systems, Neurocomputing 562 (2023). https://doi.org/10.1016/j.neucom.2023.126847.

[27] X. Zhang, S. Fan, Z. Song, Reinforcement learning-based cost-sensitive classifier for imbalanced fault classification, Sci. China Inf. Sci. 66 (2023). https://doi.org/10.1007/s11432-021-3775-4.

[28] G. Taguchi, S. Chowdhury, Y. Wu, The Mahalanobis-Taguchi System (2001).

[29] Q. Chen, F. Nie, W. Yu, X. Li, l_2,p-Norm and Mahalanobis distance-based robust fuzzy C-Means, IEEE Trans. Fuzzy Syst. 31 (2023) 2904–2916. https://doi.org/10.1109/TFUZZ.2023.3235384.

[30] V.B. Mukkanti, A.R. Tembhurkar, Taguchi's optimization technique for water deflouridation through electrocoagulation, J Dispers Sci Technol (2023). https://doi.org/10.1080/01932691.2023.2289622.

[31] H. Qian, Y. Yu, Derivative-free reinforcement learning: a review, Front Comput Sci 15 (2021). https://doi.org/10.1007/s11704-020-0241-4.

[32] A. Sadollah, H. Sayyaadi, A. Yadav, A dynamic metaheuristic optimization model inspired by biological nervous systems: Neural network algorithm, Appl Soft Comput 71 (2018) 747–782. https://doi.org/10.1016/j.asoc.2018.07.039.

[33] E. Mortaz, Imbalance accuracy metric for model selection in multi-class imbalance classification problems, Knowl Based Syst 210 (2020). https://doi.org/10.1016/j.knosys.2020.106490.

[34] K.S. Gyamfi, J. Brusey, A. Hunt, E. Gaura, Linear dimensionality reduction for classification via a sequential Bayes error minimisation with an application to flow meter diagnostics, Expert Syst Appl 91 (2018) 252–262. https://doi.org/10.1016/j.eswa.2017.09.010.

[35] M. Bator, Dataset for sensorless drive diagnosis [Dataset], UCI Machine Learning Repository, 2013. https://doi.org/10.24432/C5VP5F.

[36] R. Wu, Y. Ren, M. Tan, L. Nie, Fault diagnosis of HVAC system with imbalanced data using multi-scale convolution composite neural network, Build Simul 17 (2024) 371–386. https://doi.org/10.1007/s12273-023-1086-1.