Advances in Reliability, Safety and Security, Part 2 – Kołowrocki, Magryta-Mut (eds.)
© 2024 Polish Safety and Reliability Association, Gdynia, ISBN 978-83-68136-14-2 (printed), ISBN 978-83-68136-01-2 (electronic)

Advances in Reliability, Safety and Security

ESREL 2024
Monograph Book Series

Embeddable Fault Diagnosis Method Based On Improved Fuzzy Broad Learning System

Yunxiang Long[a], Chunyue Gu[a], Shaowei Chen[a], Shuai Zhao[b]

[a]*School of Electronics And Information, Northwestern Polytechnical University, Xi'an, China*
[b]*AAU Energy, Aalborg University, Aalborg, Denmark*

Abstract

With the rapid development of industrial production processes towards large-scale, complexity, and intelligence, fault monitoring and diagnosis to eliminate potential risks are crucial in the industrial field. However, conventional data-driven fault diagnosis methods necessitate substantial historical data, and extensive computational support, lacking efficient edge deployment techniques, and are unable to achieve online model updates. In this paper, a fault diagnosis method for edge electromechanical equipment based on incremental learning and improved fuzzy broad learning system (IFBLS) is proposed, which utilizes fuzzy C-mean clustering improved by subtractive clustering to realize the improvement of the fuzzy broad learning system. The method achieves automatic hyperparameter selection and incremental learning of enhanced nodes. Meanwhile, the edge-end porting and deployment scheme is proposed to achieve broad learning system inference operations and online model on the STM32 microcontroller. Experimental validation is conducted using the NASA electromechanical actuator dataset, and the results demonstrate that the suggested fault diagnosis method achieves a high accuracy rate.

Keywords: electromechanical actuators, fault diagnosis, improved fuzzy broad learning system, edge computing

1. Introduction

Fault diagnosis is a crucial task for ensuring the safe and reliable operation of industrial processes. Efficient and accurate identification of abnormal faults is essential for optimizing the operation of these processes. As industrial processes have become more complex over the years, ensuring their safety and reliability has become increasingly challenging (Yu et al., 2020). In aerospace systems, electromechanical actuators play a crucial role as key components. As a result, intelligent fault diagnosis has gained significance in research. NASA and Impact Technologies have published a paper stating that they have created a flyable testbed at NASA's Ames Research Center (Li et al., 2009). This test bed is used to analyze data and models of both normal and faulty states of electromechanical actuators (EMAs).

To enhance the diagnostic capability of different faults, scholars have shown great interest in accurate and efficient data-driven fault diagnosis methods (Zhao et al., 2021). Machine learning technology has been widely used in intelligent industrial fault diagnosis, commonly used methods are support vector machine (SVM), random deep forest (RF), etc. (Zhao et al., 2021). In recent years, deep learning has been proposed and applied in fault diagnosis. Zhao et al. proposed a PIML-based parameter estimation method demonstrated by a case study of dc–dc Buck converter. It overcomes the challenges related to training data, accuracy, and robustness which a typical data-driven approach has (Zhao et al., 2022). Yang et al. proposed an EMA fault diagnosis method based on a convolutional neural network (CNN) and maximum mean deviation (Yang et al., 2019). However, the deep learning approach has certain shortcomings and limitations (Chen et al., 2018; Liao et al., 2023):

- It requires a significant amount of historical data to adjust the deep structure, which may be challenging to obtain in real industrial processes;
- As deep learning continues to evolve, the demand for computing power is increasing, making it difficult to deploy in the front end of industrial equipment;
- A complete retraining process is necessary to update the model for new samples or fault types.

Moreover, as the Internet of Things (IoT) rapidly advances, an increasing number of enterprises and factories are integrating their equipment with the IoT, which is guiding another paradigm shift in manufacturing(Zhang et al., 2023). Consequently, the conventional fault monitoring and diagnostic system is no longer satisfy the requirement of modern industrial production and manufacturing. Edge computing is now being used in intelligent industrial fault diagnosis. The edge device refers to a computational model that operates closer to the equipment's front end, away from the center of the Internet. This new model allows for tight integration with industrial equipment, resulting in enhanced real-time monitoring and faster response to faults (Shi et al., 2016). Crocioni et al. conducted a study on the lifespan of lithium batteries. They developed and trained a neural network model on a PC, which was later transferred to an STM32 MCU (Crocioni et al., 2020). Zhang et al. optimized the neural network architecture of a 2D neural network (CNN2D) and successfully implemented it on an STM32 MCU for the recognition and classification of bearing fault vibration signals (Zhang et al., 2022). However, due to the high computation cost of deep learning models and the resource constraint of mobile and embedded devices, it is hard to directly deploy deep learning models on mobile and embedded devices.(Chen et al., 2020) The most cutting-edge issue in the current academic community revolves around deploying fault diagnosis algorithms on edge devices, particularly MCU devices.

The Broad Learning System (BLS) is an efficient incremental learning system that does not require a deep architecture (Chen et al., 2018). It is designed to solve problems in deep learning. BLS is designed to utilize the mapping features of input data as inputs to a random vector functional link neural network(RVFLNN). This approach offers a simple structure, fast training, high accuracy, and incremental learning (Ren et al., 2021). BLS provides a new method for intelligent fault diagnosis at the edge end.

This paper presents an improved fuzzy broad learning system in fault diagnosis. The main contributions of this research are as follows:

- The fuzzy C-mean clustering technique is improved by incorporating subtractive clustering, resulting in an improved fuzzy broad learning system. This technique enhances the accuracy of fault diagnosis by optimizing the hyperparameter configuration approach;
- The improved fuzzy broad learning system preserves the traits of traditional BLS while enabling incremental learning of improved nodes. This allows for dynamic adjustment of the model structure, thereby enhancing the accuracy of fault diagnosis;
- A deployment scheme is proposed to implement the broad learning system on an STM32 microcontroller, enabling intelligent fault diagnosis at the edge devices.

The rest of this article is organized as follows: Section 2 presents BLS, FBLS, and IFBLS. Section 3 explains the deployment method of IFBLS on STM32. Section 4 analyzes the performance of IFBLS on PC and STM32 using the NASA electromechanical actuator dataset. Lastly, Section 5 offers discussion and conclusions.

2. Improved Fuzzy Broad Learning System

2.1. Broad Learning System (BLS)

Chen et al. proposed a width learning system based on a random vector function connected network (Chen et al., 2018). The hidden layer of BLS consists of feature nodes and enhanced nodes, as shown in Figure 1.

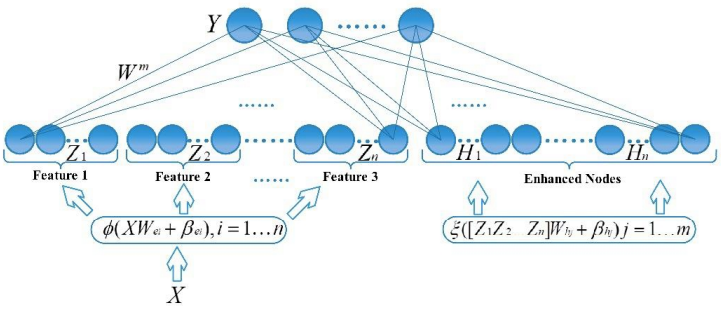

Fig. 1. Broad Learning System Structure Diagram.

Firstly, the features mapped by the input data X are used as the feature nodes of the network.

$$Z_i = \phi_i(XW_{ei} + \beta_{hi}). \quad (1)$$

where $i = 1 \ldots n$, ϕ_i are the activation functions, W_{ei}, β_{hi} are the random weights and biases, respectively, which are fine-tuned by the sparse self-encoder. The feature layer of the n group of feature nodes is denoted as $Z^n = [Z_1, Z_2, \cdots, Z_n]$.

Second, the mapped features are augmented into enhanced nodes with randomly generated weights。

$$H_j = \xi_j(Z^n W_{hj} + \beta_{hj}). \quad (2)$$

where $j = 1 \ldots m$, ξ_j are the activation functions, W_{hj}, β_{hj} are the random weights and biases, respectively, and the enhanced layer of the m group of enhanced nodes is denoted as $H^m = [H_1, H_2, \cdots, H_m]$.

Finally, all mapped features and enhanced nodes are directly connected to the output and the BLS output is

$$Y = [Z^n | H^m] Wm. \quad (3)$$

where W^m is the weight of the hidden layer to the output layer, let $A = [Z^n | H^m]$, the above equation is simplified as $Y = AW$, and the pseudo-inverse ridge regression technique is used to solve for the desired connection weights (Huang et al., 2022).

$$W = (\lambda I + AA^T)^{-1} A^T Y. \quad (4)$$

where is λ the regularization factor.

2.2. Fuzzy Broad Learning System (FBLS)

Fuzzy Broad Learning System (FBLS) is a hybrid of the Takagi-Sugeno fuzzy system() and the traditional BLS, which leverages fuzzy logic theory to learn fuzzy rules (Feng et al., 2020). This approach maintains the fast computational properties of traditional BLS while also improving its ability to handle various fuzzy logic problems. Additionally, the fuzzy rules learned by FBLS are interpretable. The structure of Fuzzy-BLS is depicted in Figure 2. (Zhou et al., 2021).

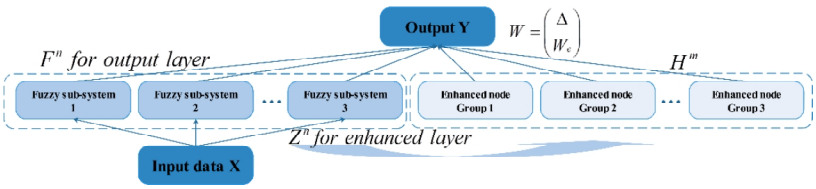

Fig. 2. Fuzzy Broad Learning System Structure Diagram.

FBLS replaces the feature nodes in the traditional BLS with fuzzy subsystems. The structure of each fuzzy subsystem is shown in Figure 3. TS fuzzy system, a commonly used fuzzy model, has been widely applied in different fields, including nonlinear system modeling, identification, fuzzy control, and fuzzy inference (Li et al., 2013). In Takagi-Sugeno fuzzy system, let $X = (x_1, x_2, \ldots \ldots, x_m)$, $x_s = (x_{s1}, x_{s2}, \ldots \ldots, x_{sm})$, there exists $IF - Then$ rule, i.e. $IF\ x_{s1}$ is A^i_{k1} and x_{s2} is A^i_{k2}…… and x_{sm} is A^i_{km}, $Then\ Z^i_{sk} = f^i_k(x_{s1}, x_{s2}, \ldots \ldots, x_{sm})$,i.e.

$$Z^i_{sk} = f^i_k(x_{s1}, x_{s2}, \ldots \ldots, x_{sm}) = \sum_{t=1}^{m} \alpha^i_{kt}\, x_{st}. \quad (5)$$

where α^i_{kt} is the coefficient. Then, the ith fuzzy subsystem of the training sample x_s can be expressed as

$$Z_{si} = \left(w^i_{s1} Z^i_{s1}, w^i_{s2} Z^i_{s2}, \ldots \ldots w^i_{sk} Z^i_{sk}\right). \quad (6)$$

w^i_{sk} can be found by the following formula.

$$w^i_{sk} = \frac{\tau^i_{sk}}{\sum_{k=1}^{ki} \tau^i_{sk}}, \quad (7)$$

$$\tau^i_{sk} = \prod_{t=1}^{m} \mu^i_{kt}(x_{st}), \quad (8)$$

$$\mu^i_{kt}(x) = e^{-\left(\frac{x - c^i_{kt}}{\sigma^i_{kt}}\right)^2}. \quad (9)$$

where σ_{kt}^{i} is taken as 1 and c_{kt}^{i} is directly obtained by $k-means$ algorithm. The output vector of the ith fuzzy subsystem for all samples is denoted as $Z_i = [Z_{1i}, Z_{2i}, \cdots Z_{si}]$. Then the output of n fuzzy subsystems is $Z^n = [Z_1, Z_2, \cdots, Z_n]$.

The outputs of the fuzzy rules generated by each fuzzy subsystem are not immediately summarized into a single value but are all sent to the enhanced layer for further nonlinear transformations to get the

$$H_j = \xi_j\left(Z^n W_{hj} + \beta_{hj}\right). \tag{10}$$

The enhanced layer of the m group of enhanced nodes is denoted as $H^m = [H_1, H_2, \cdots, H_m]$. To preserve the input characteristics, the defuzzification output of all fuzzy subsystems.

$$F_{si} = \left(\sum_{k=1}^{ki} w_{sk}^{i}\left(\sum_{t=1}^{m} \delta_{k1}^{i}\alpha_{kt}^{i}x_{st}\right) \ldots \ldots\right). \tag{11}$$

where δ_{k1}^{i} are the coefficients, then the features layer can be expressed as

$$F_i = (F_{1i}, F_{2i}, \ldots, F_{ni}) \triangleq D\Omega^i\delta^i.$$

where $D = diag\{\sum_{t=1}^{m}\alpha_{kt}^{i}x_{1t}, \ldots, \sum_{t=1}^{m}\alpha_{kt}^{i}x_{nt}\}$. $\Omega^i = \begin{pmatrix} \omega_{11}^{i} & \cdots & \omega_{1k}^{i} \\ \vdots & \ddots & \vdots \\ \omega_{n1}^{i} & \cdots & \omega_{nk}^{i} \end{pmatrix}$. $\delta^i = \begin{pmatrix} \delta_{11}^{i} & \cdots & \delta_{kn}^{i} \\ \vdots & \ddots & \vdots \\ \delta_{1n}^{i} & \cdots & \delta_{kn}^{i} \end{pmatrix}$.

The output of n fuzzy subsystems is $F^n = (F_1, F_2, \ldots \ldots, F_n)$. The output of the enhanced layer is combined to obtain the final model output.

$$Y = F^n + H^m W_e. \tag{13}$$

where $F^n = \sum_{i=1}^{n} F_i = \sum_{i=1}^{n} D\Omega^i\delta^i \triangleq D\Omega\delta$, the final model output can be pushed (Zhou et al., 2021).

$$Y = D\Omega\delta + H^m W_e \triangleq (D\Omega, H^m)W. \tag{14}$$

The weight is

$$W = (D\Omega, H^m)^+ Y. \tag{15}$$

where $(D\Omega, H^m)^+ = ((D\Omega, H^m)^T(D\Omega, H^m))^{-1}(D\Omega, H^m)^T$.

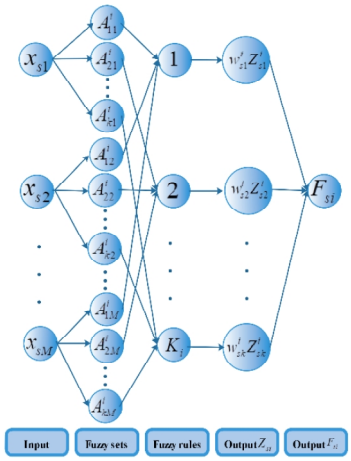

Fig. 3. Takagi-Sugeno Fuzzy System Structure Diagram.

2.3. Improved Fuzzy Broad Learning System(IFBLS)

In FBLS, K-means clustering is commonly used, along with other clustering algorithms such as fuzzy C-means clustering(FCM) and subtractive clustering(SCM). The advantage of fuzzy C-means clustering is its ability to use membership degrees to represent the extent to which data points belong to a specific class. When the initial cluster center is selected appropriately, it can achieve optimal calculation accuracy. However, there are also drawbacks.

The disadvantage lies in the number of cluster centers, which must be predetermined. In subtractive clustering, the candidate set of cluster centers is based on data points, and its computational complexity is independent of the input data dimension. Instead, it has a simple linear relationship with the number of input data points. As a result, the algorithm can calculate quickly. Nonetheless, the drawback of subtractive clustering is that the desired cluster center must be among the original data points (Wu et al., 2021). In this paper, the optimal number of cluster centers for FCM is determined through subtractive clustering. The obtained cluster centers are then used as the initial centers for FCM, improving its performance. However, a small adjustment needs to be made to the subtractive clustering centers because they are derived from the original data set. If used directly in FCM, a division by zero error may occur. By replacing the traditional K-means clustering with the improved FCM, the enhancement of FBLS is achieved. IFBLS enables the automatic selection of the hyperparameter for the number of cluster centers. The algorithm is shown in Algorithm 1.

Algorithm 1 Improved Fuzzy Broad Learning System

Input: training set, feature nodes, fuzzy subsystems, enhanced node groups
Output: weight matrix W

1. Initialization coefficients α_{kt}^{i} , using a uniform distribution in [0,1]
2. for i = 1 to n do
3. Apply subtractive clustering to the training set to get the number of cluster centers and the value of the cluster centers
4. Use the number of cluster centers obtained in Step 3 as the number of fuzzy C-means cluster centers, and the value of the cluster centers as the fuzzy C-means to initialize the cluster centers to obtain a new cluster center value
5. Use the obtained cluster center value initial Gaussian membership function center
6. for s = 1 to N do
7. Calculate $Z_{si},\ F_{si}$
8. end for
9. Calculate Z_i , F_i
10. end for
11. Calculate $Z^n,\ H^m,\ F^n,\ W$

2.4. Incremental learning By Adding Enhanced Nodes

When the improved fuzzy width learning system cannot achieve the desired performance after the training, it is necessary to add new enhanced nodes to reduce the loss function, let the addition of p enhanced nodes, $A^m = (D\Omega, H^m)$ and W^m are the feature and enhanced layers before the enhanced nodes are added, $A^{m+1} = [A^m|H^{m+1}] = [A^m|\xi(Z^n W_{hm+1} + \beta_{hm+1})]$ and W^{m+1} are the feature and enhanced layers after the enhanced nodes are added, and W_{hm+1} is the weights of the new enhanced nodes, then there are (Chen et al., 2018)

$$(A^{m+1})^+ = \begin{bmatrix} (A^m)^+ - db^T \\ b^T \end{bmatrix}. \tag{16}$$

where $d = (A^m)^+ \xi(Z^n W_{hm+1} + \beta_{hm+1}),\ b^T = [\begin{matrix} (c)^+ \\ (1 + d^T d)^{-1} d^T (A^m)^+ \end{matrix}],\ c = \xi(Z^n W_{hm+1} + \beta_{hm+1}) - A^m d,$ if $c \neq 0,\ b^T = (c)^+$, if $c = 0,\ b^T = (1 + d^T d)^{-1} d^T (A^m)^+$, so the new weights are updated as follows.

$$W^{m+1} = \begin{bmatrix} W^m - db^T Y \\ b^T Y \end{bmatrix}. \tag{17}$$

3. Edge Device Implementation

Neural networks for implementing optimization on embedded devices have gained significant attention from both academia and industry. The typical neural network approach comprises two key components: training and inference (Ying et al., 2021). Training involves searching for suitable network parameters using predefined dataset configurations, while inference entails obtaining output results from the trained neural network using new data. Typically, the operation of neural network reasoning is primarily implemented in embedded systems. Embedded microcontroller unit (MCU) devices have higher integration than MPUs, with smaller sizes and lower power consumption. MCUs are used as edge nodes and deployed with intelligent fault diagnosis algorithms, which can

further enhance the real-time and fault response speed of monitoring systems. In this paper, an improved fuzzy broad learning system deployment scheme is applied to the edge end. The goal is to achieve intelligent fault diagnosis on an STM32 microcontroller using improved fuzzy broad learning system inference operation (Chen et al., 2021).

3.1. Model Train

Begin by initializing the system, preprocessing the data, and seeing the initial parameters. These parameters include the number of feature nodes and enhanced nodes. Choose tansig as the activation function and set the initial network structure. Model train is completed on the computer side using the MATLAB software platform for the improved fuzzy broad learning system to obtain the weights of the hidden layer to the output layer W^m and the parameters of W_{ei}, β_{ki}, W_{hj}, β_{hj}. If the fault diagnosis rate does not meet the desired performance, the model network can be improved by the incremental learning of enhanced nodes.

3.2. Functions Encapsulate

To execute the model's reasoning process at the edge end, we need to encapsulate the necessary operation processes for model reasoning into a function. This function will take the training dataset, the weights, and parameters obtained in model train as input, and provide the fault diagnosis results as output. Then use the MATLAB Coder toolbox to generate the lib library from the function packaged. Typically, the mex file is generated first, executed to verify the translation result, and then the lib library is generated.

3.3. Model optimization

To enhance the operational efficiency of the generated lib library file on STM32, optimization can also be applied to algorithms, parameters, and other aspects. For storing weight coefficients, a fixed-point quantization method can be used. This method is suitable for representing network weight coefficients using 8-bit data. In the calculation process, the multiplication method for reducing subscript calculation is used to minimize the use of multiplication. For instance, when performing the operation $Y = AW$, the weight data W is accessed in the form of a linear array, and the internal data of the weight data W is accessed in the form of a multiplication calculation array subscript during specific calculations. This multiplication can be eliminated (Ying et al., 2021).

3.4. Model transplantation

Use Keil uVision5 to create the project file, port the lib libraries to the project file, and declare them. Copy the MATLAB header functions to the project file and declare them. Complete the header file declaration, initialization, and configuration of the main program in the project file main.c. Additionally, add any other necessary functions or project modules based on specific usage requirements. Finally, compile the program and burn it to the STM32 for verification.

4. Experimental Results

4.1. Experimental Data Set

To validate the performance of the model proposed in this paper, the experimental dataset used is NASA's published electromechanical actuator dataset. NASA has developed and built a Flyable Electromechanical Actuator (FLEA) test bed for gathering data while the electromechanical actuator is in operation. The collected data includes four types of faults: normal (Normal), ball screw jam fault (Jam), ball screw spalling fault (Spall), and motor winding short circuit fault (Motor) (Yang et al., 2019; Balaban et al., 2009). Choose six attributes of the Nut X Accelerometer Axis Y, Nut X Accelerometer Axis X, Nut X Accelerometer Axis Z, Motor X Temperature, Nut Y Accelerometer Axis X, Nut Y Accelerometer Axis Y, and proceed with subsequent experiments (Chen et al., 2022).

4.2. Model Parameter

The following experiments were conducted using the MATLAB R2022a software platform on a computer with an Intel(R) Xeon(R) W-2133 CPU @ 3.60GHz and 64GB RAM.

In this section, we will focus on selecting appropriate parameters to support subsequent experiments. Within FBLS, there are three main hyperparameters: Nf, Nm, and Ne. Initially, Nf is fixed at 3 and the test accuracy is evaluated for varying values of Nm and Ne. Figure 4 illustrates the impact of Nm and Ne on training accuracy and time. It is evident that performance is consistent within a certain range of Nm and Ne values, resulting in a relatively satisfactory diagnostic outcome.

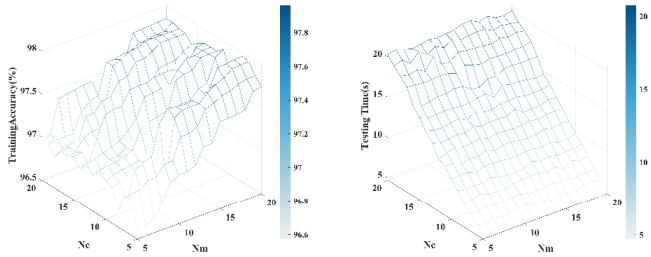

Fig. 4. Training Accuracy and Training Time of IFBLS with different Nm and Ne for the NASA electromechanical actuator dataset.

However, it is evident that as the number of nodes increases, both the accuracy rate and the training time also increase. Therefore, it is crucial to select an appropriate parameter. Traditionally, the grid search method is commonly employed for this purpose, although it is time-consuming and costly. For instance, when considering the hyperparameter value range of [0,10], grid search requires 1000 iterations to determine the optimal value. Figure 5. illustrates the effect of varying the Range of Influence of the Cluster Center on the Number of Cluster Centers, Training Accuracy, Testing Accuracy, Training Time, and Testing Time for the NASA electromechanical actuator dataset. It is evident that as the Range of Influence of the Cluster Center increases, the number of cluster centers decreases, along with a decrease in time cost. However, this increase also results in a decrease in accuracy rate, which aligns with the expected outcome.

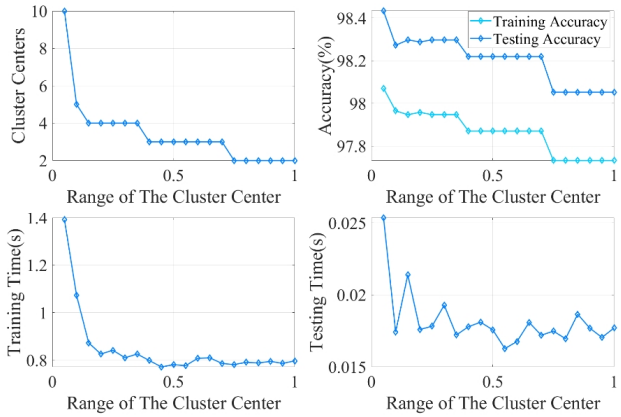

Fig. 5. The effect of varying the Range of Influence of the Cluster Center on the Number of Cluster Centers, Training Accuracy, Testing Accuracy, Training Time, and Testing Time for the NASA electromechanical actuator dataset.

4.3. Model Effect Analysis

In this section, we present the experimental results that validate the model's effectiveness. Firstly, the various clustering methods employed by FBLS are examined, as depicted in the table. The unified network structure utilized for comparison is 3-6-20. It is evident from the results that the model proposed in this paper achieves the highest accuracy rate in fault diagnosis. However, this comes at the expense of increased time consumption compared to the other methods. Nonetheless, since there is no requirement for conducting a mesh search for the

Nf hyper-parameters, a suitable network structure can be directly determined. Consequently, in practical applications, the time cost may be reduced. The result is shown in Table 1.

Table 1. Performance comparison on the NASA electromechanical actuator dataset with different clustering methods.

Model	Nf	Nm	Ne	Train Time	Train Accuracy	Test Time	Test Accuracy
K-means	3	6	20	1.7149	97.5833%	0.10945	97.2206%
FCM	3	6	20	1.9068	97.6733%	0.10591	97.4604%
SCM-FCM	3	6	20	1.9539	97.7533%	0.10459	97.7805%

Compare the model with FBLS, BLS, and SVM, as demonstrated in Table 2. The IFBLS model exhibits a notable enhancement in accuracy, while still preserving the computational efficiency of the BLS model. In comparison to SVM, the training time is significantly faster. For comparison and improvement of the FBLS model, the network structure of FBLS and BLS is standardized to 3-6-20. The SVM model utilizes the libsvm matlab toolbox for determination (Chang et al., 2011).

Table 2. Performance comparison of SVM, BLS, FBLS, and IFBLS for the NASA electromechanical actuator dataset.

Evaluation Indicators	IFBLS	FBLS	BLS	SVM
Training Time	1.9539	1.7149	1.2935	28.707
Training Accuracy	97.753%	97.5833%	95.637%	91.533%
Testing Time	0.10459	0.10945	0.0927	0.7165
Testing Accuracy	97.980%	97.2206%	95.541%	91.642%

To evaluate the incremental effect of the model-enhanced node, the initial network structure is set as 2-6-20. Each time, the enhanced nodes are dynamically increased by 20, resulting in a final structure of 2-6-100. Table 3 displays the results, indicating that the model accuracy improves to some extent as the enhanced nodes increase. This incremental approach proves effective. In cases where the improved fuzzy width learning system fails to achieve the desired performance after training, adding new enhanced nodes can enhance the diagnostic accuracy.

Table 3. Performance comparison on the NASA electromechanical actuator dataset with different numbers of enhanced nodes.

Nf	Nm	Ne	Train Accuracy	Test Accuracy
		20	97.49%	97.3405%
		40	97.8583%	97.8204%
2	6	60	97.8783%	97.8371%
		80	97.8217%	97.8805%
		100	98.145%	98.1006%

The experiments above demonstrate that an improved fuzzy broad learning system performs effectively in diagnosing faults in the NASA actuator dataset. It maintains the learning speed of the BLS method, allowing for efficient model training on a regular PC with less practice required. Additionally, compared to other methods, the accuracy of fault diagnosis has been significantly improved. Furthermore, the incremental learning approach using augmented nodes further enhances the accuracy of model diagnosis. IFBLS also enables quicker determination of a reasonable network structure, eliminating the need for grid search and other time-consuming methods. This reduces the additional time cost involved.

4.4. Edge Devices Performance Analysis

Using the porting method outlined in this paper, the enhanced model has been migrated to the STM32 microcontroller, enabling the execution of width learning inference operations on the STM32H750XBH6 microcontroller with 2MB FLASH. The program's current memory capacity is 25.6kB FLASH and 9.1kB RAM. The power needed by the STM32 to execute the fault diagnosis program is 0.0418W. The running results are as shown in. Figure 6. The findings are presented in Table IV. In comparison to the computer-side implementation, the accuracy of width learning troubleshooting at the edge remains nearly unchanged, but the running time is increased due to limited computing power.

Table 4. Performance comparison on the NASA electromechanical actuator dataset with PC and Edge Devices.

Device	PC	MCU
Testing Accuracy	96.26%	96.12%
Testing Time	0.11s	0.46 s

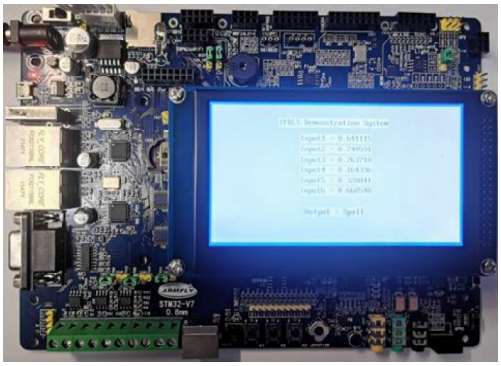

Fig. 6. The diagnosis results of the STM32 microcontroller edge terminal fault on the NASA electromechanical actuator dataset.

5. Conclusion

In this paper, we present a fault diagnosis method for edge electromechanical equipment based on an improved fuzzy broad learning system. The method is evaluated using NASA-collected data from electromechanical actuators. While preserving the benefits of a simple structure and fast training speed found in traditional BLS, we utilize fuzzy C-means clustering enhanced by subtraction clustering to address the issue of employing a grid search approach for the hyperparameter configuration of feature nodes. This enhancement aims to improve the fuzzy broad learning system and enhance the accuracy of fault diagnosis. It proposes an incremental learning approach to enhance the fuzzy broad learning system's enhanced node and further boost the model's accuracy. The improved fuzzy broad learning system enables the inference operation to be performed on the STM32 microcontroller. This allows for close integration with industrial equipment at the edge, resulting in enhanced real-time monitoring and faster response to faults.

References

Yu, W et al. 2020. Broad Convolutional Neural Network Based Industrial Process Fault Diagnosis With Incremental Learning Capability. IEEE Transactions on Industrial Electronics, 5081-5091.

Li , H et al. 2009. A new modeling reference direct adaptive sliding mode control for electromechanical actuator. Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, 6078-6082.

Zhao, S et al. 2021. Enabling Data-Driven Condition Monitoring of Power Electronic Systems With Artificial Intelligence: Concepts, Tools, and Developments. IEEE Power Electronics Magazine, 18-27.

Zhao, S et al. 2021. An Overview of Artificial Intelligence Applications for Power Electronics. IEEE Transactions on Power Electronics, 4633-4658.

Zhao, S et al. 2022. Parameter Estimation of Power Electronic Converters With Physics-Informed Machine Learning. IEEE Transactions on Power Electronics, 11567-11578.

Yang, J et al. 2019. Long short-term memory neural network based fault detection and isolation for electro-mechanical actuators. Neurocomputing, 85-96,.

Chen, J et al. 2018. Broad Learning System: An Effective and Efficient Incremental Learning System Without the Need for Deep Architecture. IEEE Transactions on Neural Networks and Learning Systems, 10-24.

Liao, X et al. 2023. Remaining useful life with self-attention assisted physics-informed neural network. Advanced Engineering Informatics, 58, 102195.

Chen,Y et al. 2020. Deep Learning on Mobile and Embedded Devices: State-of-the-art, Challenges, and Future Directions. ACM Comput. Surv. 53, 4, Article 84, 37 pages.

Shi, W et al. 2016. Edge Computing: Vision and Challenges. IEEE Internet of Things Journal, 637-646.

Crocioni, G et al. 2020. Li-Ion Batteries Parameter Estimation With Tiny Neural Networks Embedded on Intelligent IoT Microcontrollers. IEEE Access, 122135-122146.

Zhang, M et al. 2022. CNN Based Fault Diagnosis Using MCU(in Chinese). Mechanical Science and Technology for Aerospace Engineering, 1-8.
Ren,C et al. 2021. Research of broad learning system(in Chinese). Application Research of Computers, 2258-2267.
Huang, L et al. 2022. Broad Recommender System: An Efficient Nonlinear Collaborative Filtering Approach.arXiv preprint arXiv:2204.11602.
Feng, S et al. 2020. Fuzzy Broad Learning System: A Novel Neuro-Fuzzy Model for Regression and Classification. IEEE Transactions on Cybernetics, 414-424.
Zhang, C et al. 2023. A multi-access edge computing enabled framework for the construction of a knowledge-sharing intelligent machine tool swarm in Industry 4.0. Journal of Manufacturing Systems, 56-70.
Zhou, L et al. 2021. Data Analysis——Foundation, Model and Applications(in Chinese). Science Press, Beijing.
Li, G et al. 2013. Fuzzy. Predictive Control and Its Realization in MATLAB(in Chinese). Publishing House of Electronics Industry, Beijing.
Wu, X et al. 2021. Research on Applications of Improved Fuzzy System and ANFIS in Intelligent Choosing of Machining Parameters(in Chinese). Science Press, Beijing.
T. Takag et al. 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, 116-132.
Ying, R et al. 2021. AI embedded systems: Algorithm optimization and implementation(in Chinese). Machine Press, Beijing.
Chen, S et al. 2021. Development of an Online Fault-Diagnostic-System Based on STM32 for Actuators. 2021 Global Reliability and Prognostics and Health Management (PHM-Nanjing), 1-8.
Balaban, E et al. 2009. A diagnostic approach for electro-mechanical actuators in aerospace systems. 2009 IEEE Aerospace Conference, 1-13.
Chen, S et al. 2022. A Fault Diagnosis Platform of Actuators on Embedded IoT Microcontrollers. 2022 Prognostics and Health Management Conference, 210-217.
Chang, C et al. 2011. LIBSVM: A library for support vector machines. ACM transactions on intelligent systems and technology (TIST), 1-27.