Quantitative Evaluation for Robustness of Intelligent Fault Diagnosis Algorithms Based on Self-attention Mechanism

He Liu[1,2], *Cheng Wei*[1*], *Bo Sun*[2]

[1] *School of Astronautics, Harbin Institute of Technology, China*
[2] *Beijing Institute of Spacecraft System Engineering, China Academy of Space Technology, China*
49951272@qq.com, weicheng@hit.edu.cn, Sunboo2002@126.com

Abstract

Currently, various algorithmic models encounter numerous challenges in practical applications, such as noise, interference and input changes, which can significantly impact their performance. Many methods have been proposed to enhance model robustness. However, to assess the effectiveness of these improvements, it is generally necessary to compare the model's performance before and after applying the same noise and analyze the resulting changes. Moreover, to evaluate the robustness of multiple models that meet basic requirements for a specific task, a qualitative analysis is performed using specific indicators. This is especially crucial in fault diagnosis where multiple types of noise interference in the data can hinder accurate fault classification. Addressing this situation, this paper presents a quantitative evaluation method for the robustness of intelligent fault diagnosis algorithms based on the self-attention mechanism. The proposed method entails dividing the dataset into sub-datasets according to signal-to-noise ratio after injecting noise, separately calculating sub-indicators after training, dynamically assigning weights to these indicators using the self-attention mechanism and combining the weights of different sub-indicators to generate a comprehensive evaluation value for assessing robustness. The proposed method is validated through experiments involving three models, and the results demonstrate the reliability of this quantitative calculation approach for robustness.

Keywords: Implantation noise, Sub indicators, Robustness, Self-attention

1 Introduction

With the gradual improvement of computer computing ability, machine learning and deep learning have also experienced rapid development, artificial intelligence systems based on various models are being widely applied in various fields of life [1]. However, in practical applications, these models often face various challenges, and due to the presence of various noise and interference, the performance of the models may deteriorate [2]. And the robustness of a model characterizes its ability to resist different data distributions, noise, interference and input variations, determining whether the model can maintain stable predictive capabilities in the face of such noise. In the real world, data is often imperfect, containing noise, and the environment can change at any time [3-4]. Especially in the field of fault diagnosis, there are various noises in the raw data. If the model has poor robustness, its performance will significantly degrade when faced with new data, resulting in unreliable decisions and predictions. Conversely, if the model has strong robustness and can generalize well to new data, it ensures stability and reliability in practical applications. Moreover, in the field of security, malicious actors may attempt to deceive the model [5-6], making the robustness of the model crucial to defending against adversarial attacks. Therefore, studying the robustness of models and how to improve it is essential and has become an important research direction.

Currently, there are several common methods to improve the robustness of models, including data augmentation [7], adversarial training [8-9], model regularization [10-12], and ensemble learning [13-14]. Data augmentation refers to various random transformations and expansions applied to the training data, such as adding noise or randomly adding or removing data [15], aiming to increase the diversity of training data and enable the model to better handle different types of noise and transformations. Adversarial training is a method used to defend against adversarial attacks. Adversarial attacks involve deliberately introducing small perturbations into input data to deceive the model or induce incorrect predictions. Adversarial training involves using methods like Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) to generate adversarial samples for model training [16-17]. After multiple iterations of training, the model gradually learns to recognize and handle these perturbations, thereby significantly improving its robustness. Model regularization involves introducing regularization terms such as dropout, L1, and L2 into the model to penalize complex models or large parameter values [18]. This encourages the model to become smoother during the learning process, suppresses its excessive sensitivity to specific data, and improves its ability to predict unknown data. Ensemble learning involves combining multiple different base models to form a larger model that captures the advantages of different models, ultimately enhancing generalization ability and robustness [19].

Clearly, a commonality among the various methods for improving robustness is the qualitative analysis or indirect

*Corresponding Author: Cheng Wei; E-mail: weicheng@hit.edu.cn

DOI: https://doi.org/10.70003/160792642024112506012

evaluation based on performance indicators such as prediction accuracy. When comparing the robustness of multiple models, it is generally done by conducting tests using the same adversarial samples and comparing their adversarial performance based on indicators such as adversarial loss and misclassification rate. Another approach is to create a robustness benchmark test set that includes various adversarial samples, noise, and data distribution changes, and evaluate the models' performance on this test set to judge the strength of their robustness [20]. Additionally, implanting different types and degrees of noise and comparing the performance of different models under such noise can be done using indicators like signal-to-noise ratio and root mean square error to assess their robustness. Indirect comparisons or qualitative analyses not only fail to differentiate minor differences between different models, making it difficult to compare them, but also hinder the monitoring of how the robustness of models changes with data distribution.

This paper proposes a robustness quantitative evaluation method for intelligent fault diagnosis models. Firstly, the dataset is divided into several subsets based on the signal-to-noise ratio during the injection of noise. This division is done to simulate various noise interference scenarios that the model may face. Then, the model is iteratively trained using these subsets, and sub-indicators are calculated. A self-attention mechanism and a classifier are utilized to dynamically assign weights to these sub-indicators. By combining the weights of different sub-indicators, a comprehensive evaluation value is generated, which provides a comprehensive representation of the model's robustness.

This study makes the following main contributions:

(1) Use noise with different signal-to-noise ratios and divide the noise data into several sub datasets according to the signal-to-noise ratio level. This can provide training data of different signal-to-noise ratio levels for the model to perform well under various signal-to-noise ratio conditions.

(2) On the basis of the traditional self-attention mechanism, using sigmoid instead of softmax reduces the problem of gradient vanishing, making the model easier to converge when dealing with long sequences or large-scale data.

(3) A method was proposed to quantitatively evaluate the robustness of satellite power system fault diagnosis models by dynamically assigning weights to sub indicators of the dataset, and its effectiveness was demonstrated.

The remaining arrangement of this paper is as follows: in the second part, the method used in this paper is described in detail. Firstly, the method of injecting noise is introduced, followed by an introduction and analysis of the selected sub-indicators. Then, the process of generating model weights is explained, and finally, the computation of robustness is elaborated upon. The third part provides a detailed introduction to the dataset used in this paper. In the fourth part, the experimental results are presented, along with relevant analysis. The fifth part concludes the entire content of the paper and provides an outlook for future work.

2 Relevant Methodologies and Works

2.1 Overall Framework of the Method

The method adopted in this paper is shown in Figure 1 and can be divided into the following steps:

(1) Noise injection: Select random noise, periodic noise, and missing noise as three different types of noise. Among them, periodic noise selects sinusoidal signals. For each type of noise, process it using a defined noise generation function and generate noise with different signal-to-noise ratios. Then, divide the noisy data into several subsets based on different signal-to-noise ratios.

(2) Model training and sub-indicators calculation: Train the model using the subsets of data and calculate several sub-indicators. Next, compute the variance, kurtosis, and skewness of the sub-indicator data to analyze the distribution of the sub-indicators.

(3) Weight generation: Introduce the concept of self-attention mechanism to learn the importance of different sub-indicators for robustness and dynamically assign weights to them.

(4) Calculation of robustness: Multiply each sub-indicator by its corresponding weight and sum the results. The final value obtained from this computation serves as the evaluation indicator for the model's robustness.

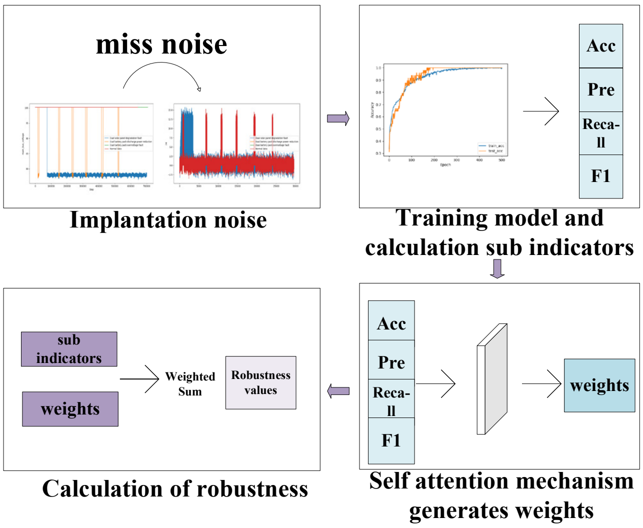


Figure 1. Method framework diagram

2.2 Noise Implantation Algorithm

This paper selects three different types of noise, namely random noise, periodic noise, and missing noise, to be inserted into the original data. When implanting noise, by defining relevant noise implantation functions, the corresponding noise variance is calculated based on different signal-to-noise ratios (SNRs), and then the number of missing data points is calculated using the length and probability of the original data. Finally, the generated noise data will be implanted into different columns of the original data. Compared to traditional methods of noise insertion, this study utilizes the signal-to-noise ratio as an indicator when introducing noise. By specifying the desired signal-to-noise ratio level, the relative strength of the signal and noise can be accurately controlled, thereby achieving precise control over the process of adding noise. The signal-to-noise ratio is used to measure the quality of a signal and represents the relative strength or power ratio between the signal and noise [21]. Its calculation formula is as follows,

$$NSR = 10 * \log \frac{Signal\,Power}{Noise\,Power} \tag{1}$$

where *Signal Power* represents the power of the signal, and *Noise Power* represents the power of the noise. From the formula, it can be observed that a smaller signal-to-noise ratio indicates a higher intensity of noise relative to the signal, meaning the signal is weaker or masked by the noise. Conversely, a higher signal-to-noise ratio indicates a clearer and higher-quality signal.

By defining three noise functions, it can control the signal-to-noise ratio range of each type of noise to be between 0-40dB and generate the desired noise data. 40dB indicates that the strength of the signal is 10000 times that of the noise, which is relatively weak and has very little impact on the signal. In this case, the signal is easily detected and recognized, and after being greater than 40dB, the obtained data is almost indistinguishable from the original data. Therefore, the maximum signal-to-noise ratio for selecting noise is 40dB. After obtaining the three types of noise, each type is divided into 40 sub-files, storing the noise data for 40 different signal-to-noise ratio levels. Then by dividing the dataset into 40 sub datasets according to the signal-to-noise ratio level, it is easy to observe the performance of the model under different noise levels and calculate the relevant sub indicators. This approach allows for the selection of data with different signal-to-noise ratio levels, enabling effective evaluation and testing of algorithms or models' performance. It is particularly useful for testing the system's robustness and accuracy under different signal-to-noise ratio conditions. Importantly, using noise data with different signal-to-noise ratios as individual training sets can significantly improve the model's robustness.

2.3 Sub-indicators and Their Definitions

After dividing the noise into 40 sub-files, the model is iteratively trained using the sub-datasets, and corresponding sub-indicators are calculated. These sub-indicators reflect the classification performance of the model in the sub-datasets, including accuracy, precision, recall, and so on. Accuracy is a performance measure of classification models, it represents dividing the number of correctly predicted samples by the total number of samples. The formula is as follows,

$$accuracy = \frac{n_{correct}}{n_{total}} \tag{2}$$

$n_{correct}$ represents the number of samples correctly predicted, and n_{total} represents the total number of predicted samples. Precision represents the proportion of true positive samples among the samples predicted as positive, measuring the accuracy of the model in predicting positive cases. The calculation formula is as follows,

$$precision = \frac{TP}{TP + FP} \tag{3}$$

where *TP* represents the number of true positive samples and *FP* represents the number of false positive samples. Recall, also known as sensitivity or true positive rate, represents the proportion of true positive samples that are successfully predicted as positive by the model. It measures the model's ability to detect positive cases. The calculation formula is as follows,

$$recall = \frac{TP}{TP + FN} \tag{4}$$

where *FN* represents the number of false negative samples. The F1 score is a weighted harmonic mean of precision and recall, used to comprehensively assess the accuracy and recall performance of the model. The calculation formula is as follows,

$$F1 = \frac{2 * TP}{2 * TP + FP + FN} \tag{5}$$

By calculating evaluation indicators for each sub-dataset, it can firstly understand the performance of the model under different noise levels and determine its robustness and generalization ability in different noise conditions. Secondly, by evaluating the model's performance on different sub-datasets, we can identify the signal-to-noise ratio levels where the model performs poorly and further improve the training strategy. After calculating all the sub-indicators, the sub-indicators corresponding to each noise type are combined in a certain order to form a vector as a data sample, and a label is assigned to it. The labels for random noise, periodic noise, and missing noise are set as 0, 1, and 2 respectively. After storing all sub-indicators by group, the variance, kurtosis, and skewness are computed for each group's evaluation indicators. Variance measures the degree of dispersion of values in the dataset [22]. The calculation formula is as follows,

$$\sigma^2 = \frac{\Sigma\left(X - \bar{\mu}\right)^2}{N} \tag{6}$$

where X is the variable. $\bar{u}$ is the population mean, and N is the sample size. Kurtosis describes the sharpness or flatness of the data distribution and is defined as the ratio of the fourth central moment to the square of the standard deviation [23]. The calculation formula is as follows,

$$Kurt\left(X\right) = \frac{E\left\{\left[X - E\left(X\right)\right]^4\right\}}{\left[\left(X - E\left(X\right)\right)^2\right]^2 - 3} \tag{7}$$

where X represents the data sample and $E(X)$ represents the expected value. Kurtosis mainly reflects the tail characteristics of data distribution. If the kurtosis is zero, it means that the data distribution is of average kurtosis; if the kurtosis is positive, it indicates that the data distribution has a peak kurtosis. And if the kurtosis is negative, it indicates that the data distribution is of flat kurtosis. Skewness, on the other hand, describes the asymmetry of the data distribution and is defined as the ratio of the third central moment to the cube of the standard deviation [24]. The calculation formula is as follows,

$$Skew(X) = \frac{E\left[\left(X - E(X)\right)^3\right]}{\left(E\left[\left(X - E(X)\right)^2\right] * \frac{3}{2}\right)} \tag{8}$$

where X represents the data sample and $E(X)$ represents the expected value. Positive skewness indicates a right-skewed data distribution, while negative skewness indicates a left-skewed data distribution. A skewness of zero indicates a generally symmetric distribution.

By calculating the variance, kurtosis, and skewness of the sub-indicators, we gain a better understanding of the stability of the model's robustness performance, its distribution characteristics, and the performance variations across different sub-datasets.

2.4 Weight Generation

This paper introduces the idea of self-attention into the model. The principle of the self-attention mechanism can be divided into the following steps, as shown in Figure 2.

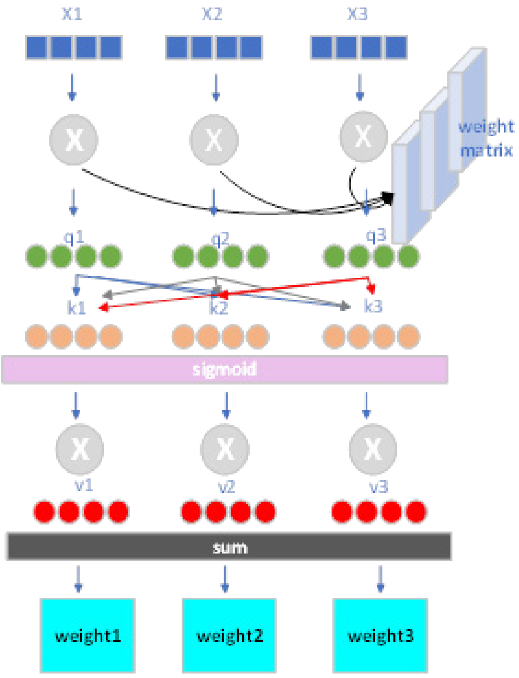


Figure 2. Schematic diagram of self-attention thought generating weights

Firstly, linear transformations are performed on each element of an input sequence to obtain the corresponding Query, Key, and Value vectors. This results in three matrices, W^q, W^k and W^v. Next, the similarity between each Query and all Keys is calculated using the following formula,

$$M = W^q \cdot \left(W^k\right)^T \tag{9}$$

In the above equation, M represents the similarity matrix. Next, the similarity matrix is normalized using functions such as softmax or sigmoid, resulting in a weight matrix M^{weight}, where each value represents a weight coefficient that reflects the similarity between Query and Key, indicating the relationships between elements. Finally, the weighted sum of the Value vectors is calculated based on the weight coefficients using the following formula,

$$N = M^{weight} \cdot W^V \tag{10}$$

In the equation, N represents the output of the self-attention mechanism. Through this process, the self-attention mechanism can focus on different parts of the input sequence at different locations, by assigning different weights to distinguish the differences in importance between different positions, and allowing the model to focus on the most important parts. Additionally, it can effectively capture long-range dependencies in the input sequence, not just limited to local information [25].

In terms of evaluating robustness, the introduction of self-attention allows the model to learn the correlations between each sub-indicator, enabling it to allocate appropriate weights to different sub-indicators. This helps to better reflect their relative importance under different noise conditions, leading to a more accurate evaluation of the model's robustness.

First, define a self-attention model composed of multiple self-attention layers, where each layer consists of the Query, Key, and Value components that work together to capture dependencies in the sequence data. The input to the self-attention model is a sequence, and each element of the sequence has an embedding vector representing its features or content. In each self-attention layer, each Query vector undergoes dot product operations with all Key vectors to obtain attention weights. Unlike traditional self-attention mechanisms, this paper chooses to use the sigmoid function instead of the softmax function. This means that by using sigmoid, attention weights are not strictly normalized, but can be taken as any value between 0 and 1, and each Key has the opportunity to be given a different weight. In this way, the model can choose the Key to focus on more freely in calculations, making it easier for the model to converge. To obtain optimal weights, the backpropagation algorithm is utilized to adjust the model parameters based on the loss function. Through multiple iterations of training, the self-attention model continuously learns and gradually optimizes the weight allocation to obtain the best weights, which are then preserved.

This paper utilizes the self-attention idea to assign weights to the variances, kurtosis, and skewness of multiple sub-indicators. This is done to determine the varying importance of each sub-indicator in assessing the model's robustness.

2.5 Calculation of Robustness

After obtaining the optimal attention weights from the self-attention section mentioned above, the weights are then allocated to the sub-indicators in the test set. The values of different sub-indicators are multiplied by their corresponding weights and summed up. This yields a quantitative evaluation of the robustness. Clearly, this value reflects the performance fluctuations of the model when facing different interferences, noises, or variations. The closer the value is to 0, the better the stability of the model. Importantly, this can be directly used to compare the strength of robustness among different models.

3 Introduction to Dataset

This paper uses four datasets, which represent normal data, degradation faults in dual solar panels, decreased discharge power in dual battery packs, and overvoltage faults in dual battery packs. These datasets are sourced from experimental data of spacecraft power system simulation models, and the model mainly include solar cell arrays, batteries, and power control units. By monitoring the parameter changes of modules such as battery discharge regulator (BDR) and battery charging regulator (BCR), various data can be obtained.

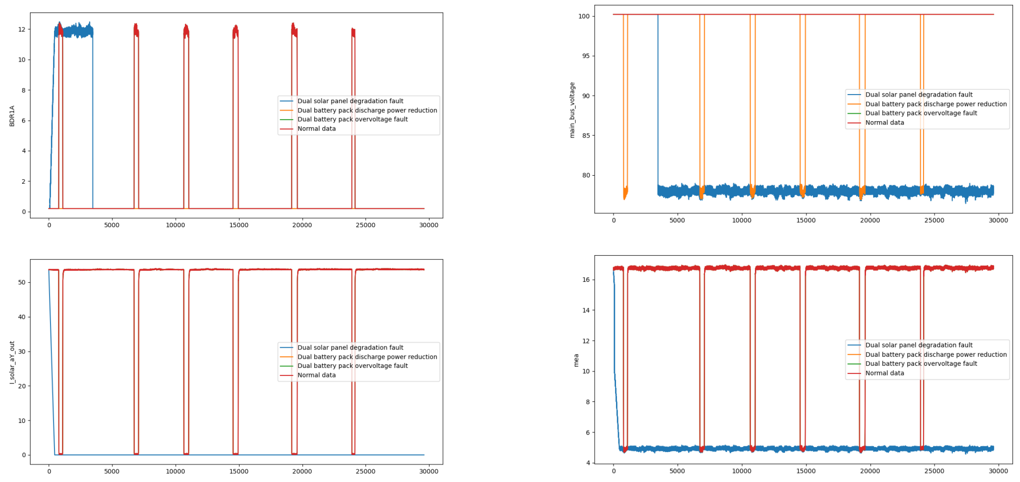
Each dataset includes 33 different indicators, such as step, current output of solar panel group aY and bY, load current of battery pack, main bus voltage, charging and discharging status of the battery pack, and the operating mode of the battery pack. The following Figure 3 to Figure 6 show the line graphs of any selected indicators in the four datasets. The blue curve represents the dual solar panel degradation fault, the orange curve signifies dual battery pack discharge power reduction, the green curve denotes dual battery pack overvoltage fault, and the red curve corresponds to normal data. It is evident from the figure that the differences between normal data and faulty data can be clearly observed.

Figure 3 depicts the variation trends of four parameters, namely BDR1A, I_solar_aY_out, main_bus_voltage, and mea, individually plotted based on the raw data. After introducing three types of noise into the original data, significant changes are observed. Taking the signal-to-noise ratio (SNR) of 10dB as an example, the following figures present the line graphs of selected data changes in the four datasets after the introduction of noise. In Figure 4, trend charts for step, I_solar_aY_out, main_bus_voltage, and VNA2 are individually presented. Upon comparing the trends of I_solar_aY_out and main_bus_voltage with those in the previous figure, it is evident that the introduction of random noise has resulted in significant fluctuations in the data.

In Figure 5, trend charts for step, I_solar_aY_out, VNA2, and main_bus_voltage are individually depicted. Following the introduction of sinusoidal periodic noise, notable alterations in both the frequency and amplitude of the data trends are observed when compared to the data without implanted noise.

In Figure 6, trend charts for step, I_solar_aY_out, BDR1A, and VNA2 are individually delineated. Following the introduction of missing noise, it is observed that the data distribution becomes highly dense. This phenomenon arises due to the absence of certain data points, causing data that should originally be positioned differently to be displayed as adjacent points on the graph.

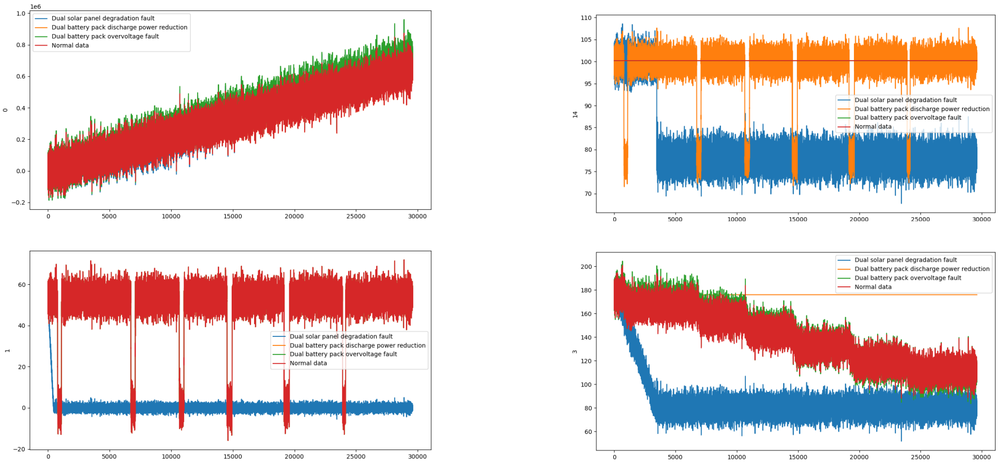
When the signal-to-noise ratio (SNR) is 10dB, it means that the power of the signal is 10 times higher than the power of the noise. At this level, the noise still poses a certain degree of interference to the signal, but this is already a relatively good signal-to-noise ratio because the signal is relatively strong and easier to identify and extract compared to noise.



(a) BDR1A and I_solar_aY_out in the original data

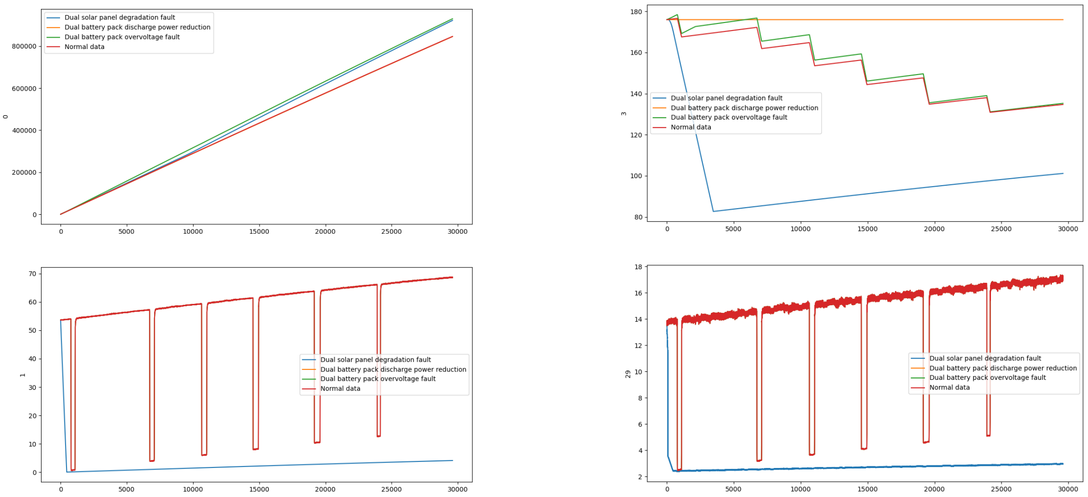
(b) main_bus_voltage and mea in the original data

Figure 3. Changes of four indicators in the original data with step



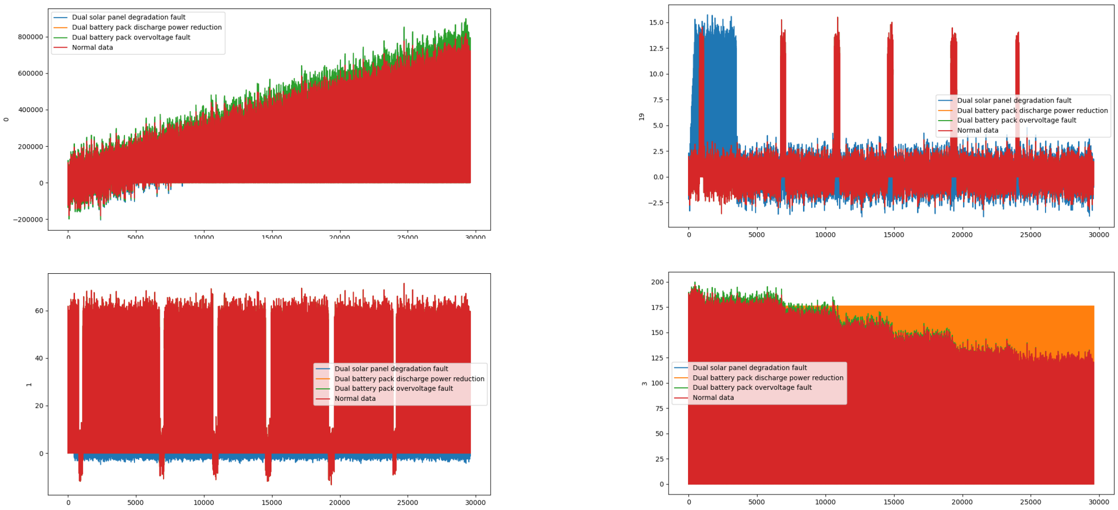
(a) Step and I_solar_aY_out in the data after implanting random noise (b) Main_bus_voltage and VNA2 in the data after implanting random noise

Figure 4. Changes of four indicators after implantation of random noise



(a) Step and I_solar_aY_out in the data after implanting periodic noise (b) VNA2 and main_bus_voltage in the data after implanting periodic noise

Figure 5. Changes of four indicators after implantation of periodic noise



(a) Step and I_solar_aY_out in the data after implanting missing noise (b) BDR1A and VNA2 in the data after implanting missing noise

Figure 6. Changes of four indicators after implantation of missing noise

4 Experiment Results and Analysis

In order to validate the reliability of the robustness quantitative evaluation indicators proposed in this paper, three models were selected for comparative experiments. The three models selected in this paper are all trained based on the deep learning framework Pytorch on the Windows 11 operating system of GEFORCE GTX1650. The training configuration parameters are shown in Table 1.

Table 1. Main parameters for model training configuration

Model	Pre training (epoch)	Generate optimal weights (epoch)	Learning rate
Model 1	50	500	0.001
Model 2	50	500	0.001
Model 3	50	500	0.001

First, selected a convolutional neural network (CNN) model named ‘model-1’ for a fault diagnosis task. The model selects ResNet-18, which is a CNN architecture within the ResNet family. ResNet-18 is a relatively shallow CNN composed of 18 convolutional layers, using Basic Blocks as the main building blocks. These basic blocks include convolutional layers, batch normalization layers, ReLU activation functions, and residual connections. It introduces residual connections on the basis of CNN to solve the problems of gradient vanishing and gradient explosion in deep network training. Residual connection is one of the key innovations of ResNet, allowing for skipping a certain number of convolutional layers, making it easier for the network to learn identity maps and better train deep networks [26]. After normalizing the original dataset, conduct model training. After training on the original dataset, predictions are made using data with injected noise, resulting in the corresponding sub-indicator dataset. A self-attention mechanism model with a feedforward neural network is used to train and test the sub-indicator data, and the weights corresponding to the best model are saved. Since the generation of these weights is influenced by the dataset partition, it even affects the final calculation results. Therefore, multiple weight generations are performed, and evaluation indicators are calculated according to the method proposed in this paper, yielding results as shown in the Figure 7.

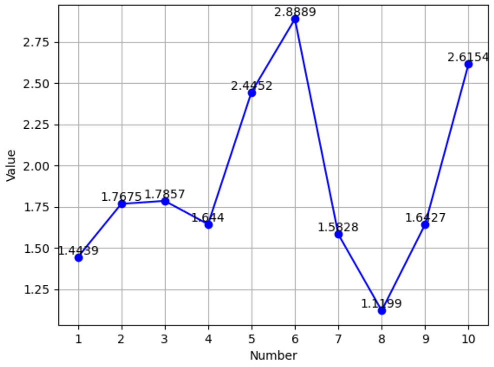


Figure 7. Evaluation indicator result chart (model 1)

Building upon model-1, model-2 is introduced with L2 regularization, which involves adding a term proportional to the L2 norm of the model parameters in the loss function. This constrains the magnitude of the model parameters and prevents overfitting, thus enhancing the model’s generalization ability and robustness [27]. After undergoing the same steps, evaluation indicator results are computed multiple times, as shown in Figure 8.

Building upon model 1, this paper introduces two changes. Firstly, during the training of the model on raw data, no normalization is performed. Normalization scales all features to the same range, aiding the model in better recognizing the relative importance of different features. Not normalizing the data may lead to the model overly relying on features with large scales, thereby reducing the model’s robustness. Secondly, the model structure is modified by using a shallower neural network with a reduced number of neurons, resulting in model 3. This also can be understood as modifying the structure of the feature extraction layer based on model 1. After following the same steps and performing multiple iterations, the results are as shown in the Figure 9.

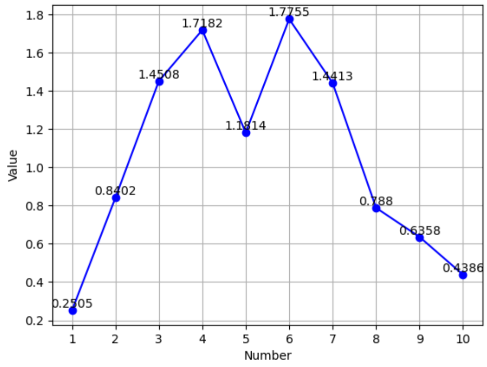


Figure 8. Evaluation indicator result chart (model 2)

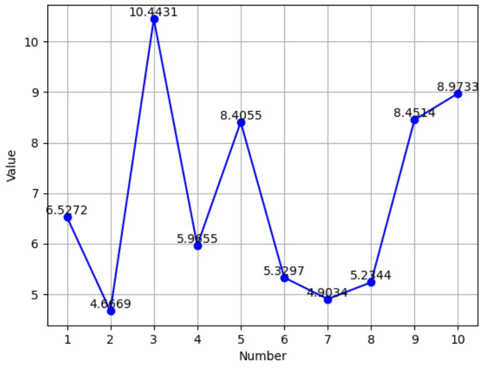


Figure 9. Evaluation indicator result chart (model 3)

Despite the fact that the best-generated weights fluctuate, resulting in fluctuating evaluation indicators calculated by the method proposed in this paper, it can be observed through multiple calculations that these indicators fluctuate within a small range and do not significantly impact the robustness comparison between models. Taking the average of the above indicators yields robustness indicators for the three models, as is shown in Figure 10, which are 1.8936, 1.0520, and 6.8900, respectively.

Clearly, the ranking of the three models' robustness corresponds with the evaluation indicators calculated using the method proposed in this paper. This demonstrates the effectiveness of this method for quantitatively assessing the robustness of models.

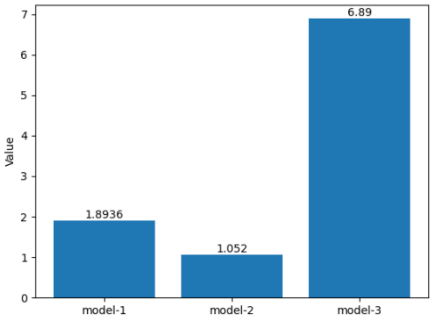


Figure 10. Numerical comparison of robustness of three models

5 Conclusion

This paper studies the robustness of intelligent fault diagnosis models and proposes a quantitative method for analyzing their robustness. Firstly, noise is implanted into the raw data and the processed data are divided into multiple sub datasets based on signal-to-noise ratio levels. Then, corresponding sub-indicators are calculated through iterative training. The idea of self-attention mechanism is introduced to analyze these sub-indicators and dynamically assign weights. Finally, the weights are multiplied by the corresponding values and summed, yielding a final numerical value. This value comprehensively considers the influence of different sub-indicators on robustness and represents the model's performance variability in the presence of noise. Thus, it serves as a quantitative evaluation indicator for model robustness. After selecting three models with different levels of robustness, this indicator is calculated for each model, and the relative sizes of the indicators align with the actual situations. Consequently, the proposed method in this paper is effective. However, this paper solely focuses on performance based on specific fault datasets. Future work will focus on exploring various combinations of data.

References

[1] R. Kamble, D. Shah, Applications of artificial intelligence in human life, *International Journal of Research–Granthaalayah*, Vol. 6, No. 6, pp. 178-188, June, 2018.

[2] J. Náplava, M. Popel, M. Straka, J. Straková, *Understanding model robustness to user-generated noisy texts*, arXiv preprint arXiv, November, 2021. https://arxiv.org/abs/2110.07428

[3] A. Kaur, A. Mantri, Evaluating the Impact of Hybridization of Vision and Sensor-Based Tracking on the Accuracy and Robustness of Virtual Reality-Based Shooting Tutor for Defense Training, *International Journal of Performability Engineering*, Vol. 19, No. 9, pp. 559-567, September, 2023.

[4] Y. Huang, L. Gong, J. Zhang, H. Fan, Y. Zhang, Review of Key Elements Identification and Robustness Analysis of Power Grid based on Complex Network Theory, *International Journal of Performability Engineering*, Vol. 18, No. 5, pp. 359-368, May, 2022.

[5] J. Xu, J. Chen, S. You, Z. Xiao, Y. Yang, J. Lu, Robustness of deep learning models on graphs: A survey, *AI Open*, Vol. 2, pp. 69-78, 2021.

[6] Y. Huang, H. Hu, C. Chen, Robustness of on-device models: Adversarial attack to deep learning models on android apps, *IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, Madrid, ES, 2021, pp. 101-110.

[7] D. Shi, Y. Ye, M. Gillwald, M. Hecht, Robustness enhancement of machine fault diagnostic models for railway applications through data augmentation, *Mechanical Systems and Signal Processing*, Vol. 164, Article No. 108217, February, 2022.

[8] H. Wang, Y. Wang, Self-ensemble adversarial training for improved robustness, *arXiv preprint arXiv*, May, 2022. https://arxiv.org/abs/2203.09678

[9] T. Bai, J. Luo, J. Zhao, B. Wen, Q. Wang, Recent advances in adversarial training for adversarial robustness, *arXiv preprint arXiv*, April, 2021. https://arxiv.org/abs/2102.01356

[10] D. Jakubovitz, R. Giryes, improving dnn robustness to adversarial attacks using jacobian regularization, *Proceedings of the European conference on computer vision (ECCV)*, Munich, Germany, 2018, pp. 514-529.

[11] R. Fraanje, S. J. Elliott, M. Verhaegen, Robustness of the filtered-X LMS algorithm—part II: robustness enhancement by minimal regularization for norm bounded uncertainty, *IEEE transactions on signal processing*, Vol. 55, No. 8, pp. 4038-4047, August, 2007.

[12] L. Ma, L. Liang, A regularization method to improve adversarial robustness of neural networks for ECG signal classification, *Computers in biology and medicine*, Vol. 144, Article No. 105345, May, 2022.

[13] Y. Cai, X. Ning, H. Yang, Y. Wang, Ensemble-in-One: Learning Ensemble within Random Gated Networks for Enhanced Adversarial Robustness, *arXiv preprint arXiv*, March, 2021. https://arxiv.org/abs/2103.14795

[14] Y. Wu, K. H. Chow, W. Wei, L. Liu, Exploring Model Learning Heterogeneity for Boosting Ensemble Robustness, *arXiv preprint arXiv*, October, 2023. https://arxiv.org/abs/2310.02237

[15] M. Qian, I. McLoughlin, W. Quo, L. Dai, Mismatched training data enhancement for automatic recognition of children's speech using DNN-HMM, *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, Tianjin, China, 2016, pp. 1-5.

[16] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, P. Abbeel, Adversarial attacks on neural network policies, *arXiv preprint arXiv*, February, 2017. https://arxiv.org/abs/1702.02284

[17] O. Bryniarski, N. Hingun, P. Pachuca, V. Wang,

N. Carlini, Evading adversarial example detection defenses with orthogonal projected gradient descent, *arXiv preprint arXiv*, June, 2021. https://arxiv.org/abs/2106.15023

[18] A. Bietti, G. Mialon, J. Mairal, *On regularization and robustness of deep neural networks*, hal-01884632v1, 2018. https://hal.science/hal-01884632v1

[19] J. Xiao, SVM and KNN ensemble learning for traffic incident detection, *Physica A: Statistical Mechanics and its Applications*, Vol. 517, pp. 29-35, March, 2019.

[20] M. Paschali, S. Conjeti, F. Navarro, N. Navab, Generalizability vs. robustness: investigating medical imaging networks using adversarial examples, *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Part I*. Granada, Spain, 2018, pp. 493-501.

[21] D. H. Johnson, Signal-to-noise ratio, *Scholarpedia*, Vol. 1, No. 12, Article No. 2088, August, 2006.

[22] H. Scheffe, *The analysis of variance*, John Wiley & Sons, 1999.

[23] B. S. Chissom, Interpretation of the kurtosis statistic, *The American Statistician*, Vol. 24, No. 4, pp. 19-22, October, 1970.

[24] S. Yusoff, Y. B. Wah, Comparison of conventional measures of skewness and kurtosis for small sample size, *2012 International Conference on Statistics in Science, Business and Engineering (ICSSBE)*, Langkawi, Malaysia, 2012, pp. 1-6.

[25] Z. Niu, G. Zhong, H. Yu, A review on the attention mechanism of deep learning, *Neurocomputing*, Vol. 452, pp. 48-62, September, 2021.

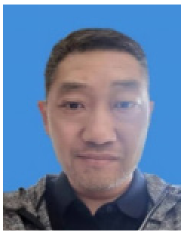
[26] Q. A. Al-Haija, M. A. Smadi, S. Zein-Sabatto, Multi-class weather classification using ResNet-18 CNN for autonomous IoT and CPS applications, *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, 2020, pp. 1586-1591.

[27] S. Singla, S. Feizi, Improved techniques for deterministic l2 robustness, *Advances in Neural Information Processing Systems 35*, New Orleans, LA, USA, 2022, pp. 16110-16124.

Biographies

He Liu received the B.S. degree from Harbin Institute of Technology (HIT), Harbin, China, in 2005, and the M.E. degree from Harbin Institute of Technology (HIT), Harbin, China, in 2007. He is currently a research fellow of Beijing Institute of Space System Engineering, CAST, Beijing, China. His current research interests include spacecraft integrated test and data processing, unattended test monitoring, spacecraft flight data simulation and analysis, machine learning, fault diagnosis.

Cheng Wei received the B.S. degree from Harbin Institute of Technology, Harbin, China, in 2005, and the Ph.D. degree from Harbin Institute of Technology (HIT), Harbin, China, in 2010. He is a professor with the School of Aerospace, Harbin Institute of Technology (HIT), Harbin, China. His current research interests include multibody system dynamics simulation, data-driven modeling and spacecraft digital twin technology.



Bo Sun received the B.S. degree from Harbin Institute of Technology, China, and the Ph.D. degree from Beihang University, Beijing, China. He is currently a professor with the Beijing Institute of Space Systems Engineering, Beijing, China. His current research interests include prognostics and health management, intelligent equipment testing, and model-driven digital twins.