



Signal Processing Technologies and Fault Diagnosis Methods Based on Edge Computing

Wenyi Guo^{1,*}

¹Southampton Ocean Engineering Joint Institute at HEU, Harbin Engineering University,
Harbin, Heilongjiang, 150001, China

*g20040428wy@hrbeu.edu.cn

Abstract. With the rapid development of the Industrial Internet of Things (IIoT) and smart manufacturing, machine signal monitoring and fault diagnosis have become critical technologies for ensuring equipment safety and operational efficiency. Traditional cloud computing-based detection methods face limitations in high data transmission volumes, significant bandwidth consumption, and poor real-time performance. Edge computing has garnered increasing attention from researchers due to its advantages in low latency, localized processing, and resource collaboration. This paper discusses the application of edge computing in machine signal processing and fault diagnosis and reviews recent key technologies and applications, with a focus on signal acquisition, preprocessing, feature extraction, and fault diagnosis methods based on lightweight deep learning models. Additionally, it analyzes data compression, real-time transmission techniques, and model deployment strategies in wireless sensor networks for rotating machinery applications. By comparing the latest research advancements, this study identifies future research directions, including edge computing platform design, algorithm lightweighting, and hardware-software co-optimization.

Keywords: Edge Computing; Industrial Internet Of Things (IIoT); Signal Processing; Fault Diagnosis; Mobilenet

1 Introduction

With the global advancement of smart manufacturing and the Industrial Internet of Things (IIoT), equipment condition monitoring and fault diagnosis have become critical for ensuring the stable operation of industrial systems [1]. Traditional cloud computing-based monitoring systems struggle to meet real-time fault response requirements due to data transmission delays and bandwidth consumption. In the early stage (1990 - 2000), standalone embedded monitoring systems primarily relied on Programmable Logic Controllers (PLCs) and microcontrollers, where sensors were directly connected to local controllers for equipment state monitoring. Although this architecture achieved millisecond-level response times, it was constrained by limited local storage capacity and computational power, making it unsuitable for multi-

© The Author(s) 2025

A. J. Moshayedi (ed.), *Proceedings of the 2025 2nd International Conference on Electrical Engineering and Intelligent Control (EEIC 2025)*, Advances in Engineering Research 279,

https://doi.org/10.2991/978-94-6463-864-6_49

dimensional data analysis under complex working conditions, let alone cross-device collaborative diagnosis. With the rise of industrial internet technologies (2000 – 2015), cloud computing-based centralized monitoring systems became prevalent. A typical example is GE's Predix platform, which transmits massive amounts of equipment data to the cloud via 5G/industrial Ethernet for big data analysis using Hadoop/Spark clusters. In recent years, edge computing has emerged as a distributed data processing architecture, moving tasks such as data processing, feature extraction, and fault diagnosis closer to edge devices near the sensors. This approach significantly reduces latency, improves real-time performance, and alleviates computational and storage burdens on cloud platforms [2].

In practical applications, machine signals—including vibration, acoustic, current, and voltage data—are often affected by noise interference, sampling frequency limitations, and data volume constraints. To address these challenges, researchers have conducted extensive studies on sensor design, signal preprocessing, feature extraction, and pattern recognition [3]. Meanwhile, breakthroughs in deep learning for image and speech recognition have led to the adoption of lightweight convolutional neural networks (CNNs) such as MobileNet for fault diagnosis on embedded edge platforms, offering new solutions for low-power, real-time diagnostics [4].

This paper discusses the application of edge computing in machine signal processing and fault diagnosis, providing a comprehensive review of existing technologies, their advantages, and limitations. The remainder of this study is structured as follows: Section 2 introduces edge computing applications in signal processing. Section 3 analyzes algorithmic and hardware optimizations to enhance edge computing efficiency. Section 4 explores future research trends and challenges in edge computing.

2 Application of Edge Computing in Signal Processing

Edge computing is a computational paradigm that deploys computing and data storage resources at the network edge, close to the data source [5]. In the context of the Internet of Things (IoT), edge computing can efficiently process massive sensor data, reduce the burden of transmitting data to the cloud, and improve system responsiveness and reliability. Its application is particularly crucial in machine signal processing. Based on the stages of signal processing, edge computing applications in this field can be categorized into signal acquisition, signal preprocessing, feature extraction, and pattern recognition.

2.1 Signal Acquisition

Signal acquisition is the first step in fault diagnosis, involving the collection of vibration, temperature, pressure, and other signals from machinery. Traditional signal acquisition systems typically rely on wired connections, which are complex to deploy and costly. However, advancements in IoT technologies have enabled the use of wireless sensor networks (WSNs) for low-cost, highly flexible machine signal acquisition. For example, the STEVAL-STWINKT1B IoT node integrates multiple

MEMS sensors capable of collecting vibration, magnetic field, and acoustic signals while performing real-time data processing at the edge [6].

Under the edge computing framework, signal acquisition devices not only gather data but also perform preliminary processing locally. To minimize data transmission, mechanical vibration wireless sensor networks (MvWSNs) have been proposed as an efficient signal acquisition method [7]. Unlike traditional wired systems, MvWSNs are self-organizing distributed systems with on-chip processing capabilities, offering advantages in flexibility, scalability, and energy efficiency.

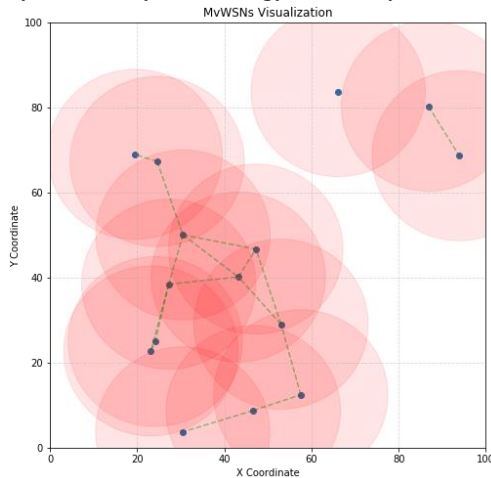


Fig. 1. Multi-hop Wireless Sensor Networks (MvWSNs) spatial topology visualization: Blue circular markers represent sensor nodes with (X, Y) coordinates spanning uniform grid. Semi-transparent red circular domains denote wireless coverage ranges, with dashed grey inter-node connections form self-organized multi-hop communication links. Two unconnected nodes at $(80, 80)$ and $(90, 90)$ suggest potential network boundary effects. (Picture credit: Original)

By deploying sensor nodes on a Drive Chain Diagnostic Simulator (DDS), vibration signals in the X and Y directions of input/output shafts were collected. This localized acquisition approach significantly reduces data transmission volume, laying the foundation for subsequent processing.

Signal acquisition optimization also involves selecting appropriate sampling frequencies and signal lengths. According to the Nyquist sampling theorem, the sampling frequency must be at least twice the highest signal frequency. However, in edge computing, excessively high sampling frequencies increase storage and computational burdens. An undersampling method can be adopted by identifying the resonance frequency band of vibration signals and configuring programmable filters, successfully reducing the sampling frequency to 1/8 of conventional methods while maintaining diagnostic accuracy.

Additionally, signal acquisition systems must consider sensor power consumption and battery life, especially in scenarios without external power sources. MvWSNs nodes employ low-power design and periodic sleep mechanisms, extending operational duration and making them suitable for long-term monitoring applications.

2.2 Signal Preprocessing

Signal preprocessing aims to enhance signal quality and reduce noise interference, providing high-quality data for feature extraction and pattern recognition. Traditional methods often perform preprocessing in the cloud, introducing transmission delays and bandwidth consumption. Edge computing addresses these issues by executing preprocessing tasks locally. For example, IIR bandpass filters and stochastic resonance algorithms can improve the signal-to-noise ratio (SNR) in real-time on edge nodes [8].

For resource-constrained MvWSNs nodes, lightweight algorithms are essential to accommodate limited computational power. Implementing signal filtering and enhancement on edge devices significantly reduces data transmission volume, alleviating communication pressure.

Another critical aspect of preprocessing is data compression to minimize transmission and storage requirements. A study by H. S. Tang et al. introduced an Opus codec-based compression method, achieving a 16:1 compression ratio for vibration signals while substantially reducing communication overhead [9]. Similarly, sparse representation-based compression algorithms on MvWSNs nodes can compress data to 0.1% of its original size while preserving key fault information. These techniques enhance transmission efficiency and prolong battery life.

Real-time performance and robustness are also vital considerations. In dynamically operating machinery, signal characteristics may vary with working conditions, necessitating adaptive preprocessing algorithms. In practice, preprocessing modules are often integrated with signal acquisition hardware on the same platform, leveraging hardware acceleration (e.g., DSP instructions) to ensure real-time processing.

Challenges in Signal Preprocessing: Algorithm Complexity vs. Hardware Limitations: Advanced methods like empirical mode decomposition (EMD) yield excellent results but are difficult to implement on edge devices. Lightweight Alternatives: Machine learning-based approaches, such as autoencoders (AE), show promise for efficient denoising under low-resource conditions, potentially advancing edge computing's preprocessing capabilities.

2.3 Feature Extraction

Feature extraction involves deriving machine state-representative parameters from preprocessed signals. While traditional methods rely heavily on expert knowledge and struggle to adapt to complex operating environments, deep learning techniques have introduced novel approaches for feature extraction. In edge computing environments, lightweight deep learning models can perform local feature extraction with high efficiency.

Recent research has systematically evaluated MobileNet architectures for this purpose. Yi et al.'s comprehensive review analyzed three generations of MobileNet models: MobileNet V1 [10], MobileNet V2 [11], MobileNet V3 [12]. Their findings revealed that while MobileNet V3 achieved superior accuracy, the original V1

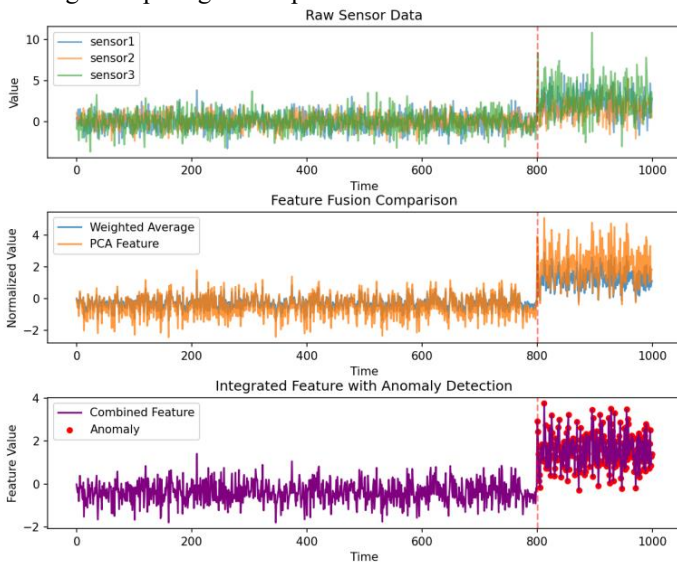
architecture remains particularly promising for MvWSNs nodes due to its simpler structure and significantly lower computational requirements.

Edge computing supports diverse feature extraction methodologies across multiple domains: (1) Time-domain features: Statistical parameters (peak values, skewness); (2) Frequency-domain features: Spectral peaks obtained through Fast Fourier Transform (FFT); (3) Time-frequency features: Short-time Fourier Transform (STFT), Wavelet Packet Transform (WPT) [13]. To address resource constraints, researchers have developed optimized approaches: Sparse representation-based methods employing dimensionality reduction techniques to decrease computational complexity [14].

Feature extraction optimization also involves multi-signal fusion. Industrial equipment typically employs multiple sensors (e.g., vibration, temperature, current), as single-signal measurements may not fully reflect machine conditions. By fusing vibration signals from X and Y directions with time-series analysis, the representational capability of features is enhanced. Edge computing enables local execution of multi-sensor data fusion, employing methods such as weighted averaging or Principal Component Analysis (PCA) to generate composite features, thereby improving diagnostic robustness.

Fig. 2 demonstrates that multi-dimensional data fusion can effectively enhance the robustness of diagnostic results. Compared to single-sensor data analysis, this approach significantly reduces both false alarm rates and missed detection rates.

Future research directions for feature extraction include adaptive feature learning and incremental learning. Adaptive feature learning enables online model updates to accommodate changing machine operating conditions, while incremental learning allows models to be progressively optimized on edge devices without requiring complete retraining. These technologies will further improve the feature extraction capabilities of edge computing in complex scenarios.



(a)

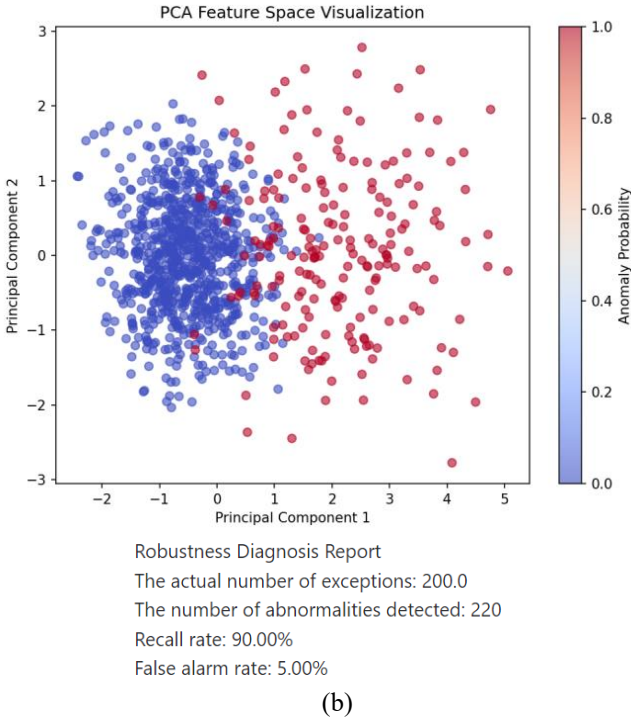


Fig. 2. (a) Multi-stage analysis: [Top] Raw temporal signals of sensor1 (blue), sensor2 (green), and sensor3 (yellow) with amplitude fluctuations (-2 to 4 units) over 1000-time units. [Middle] Feature fusion comparison between Weighted Average (blue) and PCA-derived feature (orange) showing divergence patterns. [Bottom] Integrated feature (purple) with detected anomalies (red markers) in temporal alignment. (b) Diagnostic visualization: 2D PCA projection with class-discriminative clustering, and quantitative robustness report showing high recall (90.00% of 200 actual anomalies detected in 220 predictions) with controlled false alarm rate (5.00%). (Picture credit: Original)

2.4 Pattern Recognition

Pattern recognition identifies machine states and fault types based on extracted features. Traditional methods such as Support Vector Machines (SVM) and Artificial Neural Networks (ANN) typically operate in cloud environments. However, with increasing demands for real-time performance, these approaches struggle to meet requirements.

Edge computing addresses this challenge by performing pattern recognition tasks locally, enabling rapid diagnostics. In edge computing implementations, MobileNet models are commonly deployed on MvWSNs nodes for real-time fault identification.

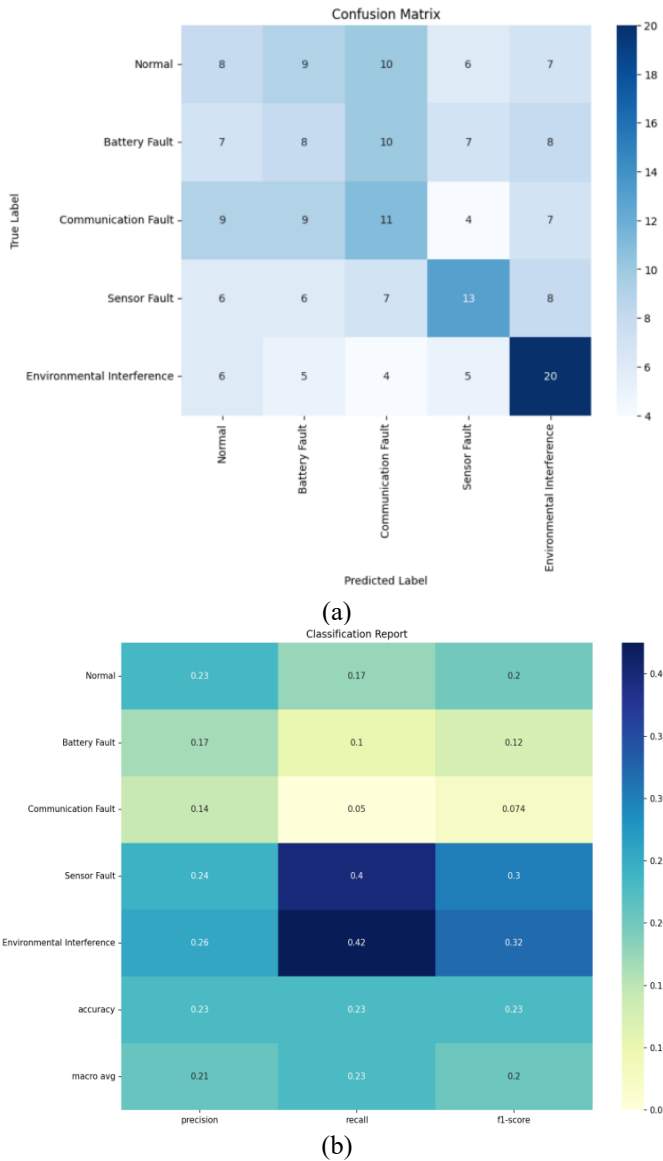


Fig. 3. (a) Confusion matrix using blue gradient coloration shows the correspondence between actual fault types (rows) and predicted labels (columns), with diagonal values indicating correct classifications. (b) Classification report heatmap displays precision, recall, and F1-score metrics using green-to-blue color scaling, with macro-average metrics and overall accuracy shown in bottom rows. (Picture credit: Original)

Yi et al. pointed out that the MobileNet model can identify multiple fault types on MvWSNs nodes with an accuracy rate of 0.98 and a calculation time of only 135ms [15].

The optimization of pattern recognition involves model selection and algorithm lightweighting. By optimizing the network structure of MobileNet, the number of parameters was reduced by approximately 70% while maintaining a high accuracy rate [15]. Although traditional machine learning methods (such as random forests) are still widely used in edge computing due to their simplicity in calculation, deep learning models perform better in complex fault recognition. To balance performance and resource consumption, hybrid models have become a research hotspot, such as combining CNN and SVM, using CNN to extract features and SVM for classification to reduce computational costs.

Edge computing supports the integration of fault diagnosis and control algorithms on the same processor, which is beneficial for anomaly detection and real-time control. This integration approach significantly improves the system's response speed.

The challenges of pattern recognition include model interpretability and real-time performance. Deep learning models are often regarded as "black boxes", and the diagnostic results are difficult to explain, which limits their application in critical tasks. Research on explainable AI technologies, such as attention mechanisms, is ongoing to reveal the basis of model decisions. In addition, the computing power of edge devices limits the deployment of complex models, and further optimization of algorithm and hardware co-design is needed.

3 Optimization of Edge Computing Efficiency

Edge computing devices typically have limited computing resources and storage space, thus requiring lightweight algorithms and optimized hardware platforms to achieve efficient fault diagnosis.

3.1 Lightweight Algorithms:

Lightweight algorithms aim to reduce model parameters and computational complexity while maintaining high diagnostic accuracy. In recent years, the rapid development of deep learning technology has provided new possibilities for lightweight algorithms, especially in edge computing scenarios.

Algorithm Design. The core of lightweight algorithms lies in reducing computational costs through structural optimization. MobileNet is a typical lightweight deep learning model that decomposes standard convolution into depthwise convolution (DW) and pointwise convolution (PW) through depthwise separable convolution (DSC) technology, significantly reducing the number of parameters and computational load. The improved MobileNet model proposed by Yi et al. further optimizes the network structure by adjusting the number of convolutional layers and channels, enabling efficient operation on resource-constrained MvWSNs nodes. Experimental results show that this model achieves an accuracy of 0.99 on the DDS dataset, with a data transmission volume of only 0.1% of the original data and a computation time of 135ms [15]. Compared with traditional convolutional neural networks (CNNs),

MobileNet reduces computational costs by approximately 9 times, providing a feasible solution for edge devices.

In addition to MobileNet, other lightweight models such as SqueezeNet and ShuffleNet are also applied in edge computing. SqueezeNet reduces the number of parameters by introducing "Fire modules", while ShuffleNet improves feature utilization through channel shuffling. These models significantly reduce computational complexity while maintaining diagnostic performance, making them suitable for resource-constrained edge devices [16,17].

Optimization Strategies. To further enhance the performance of lightweight algorithms, researchers have proposed various optimization strategies, including model pruning, quantization, and knowledge distillation.

- **Model Pruning:** By removing weights or neurons that contribute less to the output, the number of model parameters is reduced. Pruning the MobileNet model can further reduce memory usage without significantly affecting accuracy [18]. Pruning techniques are particularly suitable for edge devices, effectively reducing storage and computational requirements.
- **Quantization:** Converting model parameters from floating-point numbers to low-bit integers significantly reduces computational complexity and memory requirements. Quantization technology is widely used in edge computing, for example, deploying quantized models on STM32 microcontrollers can convert floating-point operations to integer operations, thereby improving computational efficiency [19].
- **Knowledge Distillation:** Utilizing high-performance complex models to guide lightweight models in learning, maintaining high accuracy while reducing model complexity. This method has potential in edge computing, especially in scenarios requiring high-precision diagnosis.

3.2 Specific Hardware Platforms

Specific hardware platforms provide low-power, high-performance solutions for edge computing, and their design must balance computing power, power consumption, and compatibility.

Hardware Architecture. Edge computing devices typically use low-power processors such as microcontrollers (MCUs), graphics processing units (GPUs), and application-specific integrated circuits (ASICs). Commonly used hardware platforms include the STM32 series, NVIDIA Jetson, Intel Neural Compute Stick, and Google Edge TPU [20]. These platforms each have their advantages:

- **STM32 Microcontrollers:** The STM32 series is renowned for its low power consumption and high cost-effectiveness, making it suitable for small IoT nodes. The STM32F405 microcontroller features 1MB of flash memory and 196KB of SRAM, supports floating-point operations and matrix operations, and is capable of

running lightweight deep learning models [20]. Its built-in DSP instruction set further optimizes the execution efficiency of signal processing tasks.

- NVIDIA Jetson: The Jetson series (such as Jetson AGX Orin) integrates high-performance GPUs and CPUs, making it suitable for tasks requiring parallel computing, such as deep neural network (DNN) inference. The Jetson platform supports up to 275 TOPS of computing power, making it ideal for time-sensitive fault diagnosis applications [21].
- Specialized hardware: Such as Google Edge TPU and Intel Neural Compute Stick, which are optimized for machine learning tasks and provide high-efficiency model inference capabilities. These hardware devices are widely used in edge computing, especially in industrial scenarios that require rapid response.

Development tools. The performance of hardware platforms depends on the support of efficient development tools. In recent years, hardware manufacturers have provided a variety of integrated development environments (IDEs) and toolchains, simplifying the process of algorithm deployment. The STM32Cube AI tool supports importing pre-trained models from frameworks such as TensorFlow and Keras, and automatically optimizes the models to fit STM32 microcontrollers [20]. This tool can analyze the input and output dimensions, computational complexity, and memory requirements of the model to ensure its feasibility on resource-constrained devices.

Platforms such as Microsoft Azure IoT Edge, Amazon AWS IoT, and Google Cloud IoT Core provide containerized deployment and remote management functions, making it convenient for developers to implement complex algorithms on edge devices. Additionally, open-source tools like Node-RED lower the development threshold through graphical interfaces, making them suitable for rapid prototyping and testing.

3.3 Software and Hardware Integration Platform

Rotating Machinery Fault Diagnosis Based on MobileNet. Yi et al. proposed a rotating machinery fault diagnosis method based on MobileNet, applied to MvWSNs, for real-time detection of multiple fault types in planetary gearboxes [15]. This method deploys a lightweight MobileNet model on sensor nodes and utilizes depthwise separable convolution to reduce computational complexity, achieving efficient fault diagnosis. Experiments were conducted on the Drive Chain Diagnosis Simulator (DDS), setting five states: normal, tooth surface pitting, tooth root crack, outer ring bearing fault, and inner ring bearing fault. Signals were sampled at a frequency of 2560Hz, with each sample containing 1024 points, generating a total of 800 samples, of which 600 were used for training and 200 for testing.

The experimental results show that the method achieved an accuracy of 0.99 on the DDS dataset and 0.98 on MvWSNs nodes, with a data transmission volume of only 0.1% of the original data and a computation time of 135ms. Through t-SNE visualization analysis, the feature clustering effect was good, with only a small overlap between the outer ring bearing fault (F3) and the normal state (N), indicating

strong generalization ability of the model. Additionally, this method demonstrated high real-time performance in practical tests, significantly reducing latency compared to traditional methods (transmitting raw data takes about 1000ms).

The success of this case is attributed to the lightweight design of MobileNet and the support of the STM32Cube.AI tool. After model deployment, nodes only need to transmit 5 bytes of diagnostic results, greatly alleviating network bandwidth pressure. In practical applications, this method can be extended to scenarios such as wind turbine gearboxes and high-speed rail transmission systems, achieving comprehensive monitoring of complex equipment through multi-node collaboration. However, the model's processing capability for high sampling rate signals is limited and needs further optimization to adapt to more complex working conditions.

Motor Fault Diagnosis Based on Edge Computing. Lu et al. introduced a motor fault diagnosis method based on edge computing, which significantly reduces data transmission volume by implementing signal enhancement and compression on IoT nodes [22]. This method targets motor bearing faults, using a stochastic resonance filter to enhance weak vibration signals and the Opus codec to compress signals to 1/16 of the original data, achieving a compression ratio of 16X. Experiments were conducted on the STM32WB55 microcontroller, combining BLE 5.0 protocol to transmit diagnostic results, ensuring low power consumption and efficient communication.

Specifically, the system first collects motor vibration signals through high-precision acceleration sensors at a sampling frequency of 10kHz. Subsequently, stochastic resonance filtering is performed on the edge node, dynamically adjusting filter parameters to enhance fault characteristic frequencies. The processed signals are then passed through a feature extraction module to generate time-domain and frequency-domain features, which are identified by a lightweight classifier (such as SVM) for fault type recognition. Experimental results show that this method can complete signal processing and diagnosis within 200ms, with an accuracy rate of over 95%, significantly outperforming the latency of cloud-based solutions (about 1 second).

The innovation of this case lies in the integrated design of signal enhancement and compression. The stochastic resonance filter amplifies weak signals through nonlinear processing, suitable for early fault detection; the Opus codec retains key information through adaptive compression, reducing transmission costs. In practical applications, this method has been deployed in industrial motor monitoring systems, enabling real-time health assessment of production line equipment through multi-sensor networks.

Motor fault diagnosis needs to deal with complex noise environments and multi-source interference. This can be further improved by multi-sensor fusion technology (such as joint analysis of vibration and current signals) to enhance diagnostic robustness. In the future, this method can be extended to brushless DC motors (BLDCM) or motors for new energy vehicles, and combined with edge AI to achieve more intelligent fault prediction.

4 Challenges and Future Research Trends

Although edge computing has achieved certain progress in the fields of machine signal processing and fault diagnosis, there are still many challenges that need to be addressed urgently:

4.1 Edge-Centralized and Hierarchical Architecture Optimization

As the system scale expands and application scenarios become more complex, a single edge or cloud architecture is unable to fully meet the demands for low latency, strong computing power, and massive storage. In the future, a multi-level (terminal-edge-fog-cloud) collaborative framework needs to be constructed to achieve dynamic scheduling and load balancing of computing tasks, in order to balance real-time performance and resource utilization efficiency. Research directions include elastic task allocation strategies, dynamically adjusting algorithm deployment locations and model sizes based on network bandwidth, node load, and failure risk. And Quality of Service (QoS) guarantee for different priority fault detection tasks, allocating differentiated network and computing resources for critical equipment to ensure real-time warning capabilities.

4.2 Privacy Protection and Security Defense

Industrial data often has high sensitivity, and equipment manufacturers and operators are highly concerned about the risk of data leakage. Although edge computing can reduce the transmission of raw data, model inference and updates at the node side may still expose critical information. In the future, federated learning and secure multi-party computation technologies need to be realized to achieve secure aggregation of model parameters among multiple nodes while ensuring that local data does not leave the domain. And anomaly detection and attack defense, designing lightweight tamper-proof and intrusion detection mechanisms to ensure that edge nodes can still operate stably in the event of network attacks or physical tampering.

4.3 Multi-modal and Cross-domain Sensor Fusion

Single-modal (such as vibration, acoustic, or current) diagnosis is prone to false alarms and missed detections in complex or composite fault scenarios. Future research should focus on aligning and integrating heterogeneous data, using techniques such as time series alignment and attention mechanisms, to jointly model multimodal information such as vibration, acoustic, temperature, and images. And sensor calibration and adaptive sampling, designing online calibration algorithms and sparse sampling strategies for different sensor bandwidths and dynamic ranges to reduce energy consumption and improve diagnostic robustness.

4.4 Online Learning and Adaptive Model Update

Industrial site conditions often evolve dynamically with production loads, ambient temperatures, equipment wear and tear, etc. Static models are prone to failure. It is necessary to study incremental learning and lifelong learning mechanisms to perform model fine-tuning using buffered data and pseudo-labels at the edge, enabling rapid adaptation to new fault types or operating conditions. And self-discovery of abnormal samples, automatically identifying and labeling unseen samples through unsupervised or weakly supervised algorithms to reduce manual annotation costs and improve model maintainability.

4.5 Specialized Hardware Acceleration and Energy Harvesting

Although MCUs and lightweight neural networks can largely meet some diagnostic tasks, facing higher sampling rates and more complex models, it is urgent to implement specialized neural accelerators such as Edge TPU/NPU, providing higher throughput and lower latency under limited power budgets, and combining vibration energy, electromagnetic induction, or thermoelectric power generation to achieve "zero maintenance" long-term online monitoring of nodes.

4.6 Explainability and Reliability Evaluation

In safety-critical industrial scenarios, the explainability of diagnostic results and the reliability of the system are crucial. Future research can focus on lightweight implementation of explainable AI (XAI) in edge inference, providing fault mechanisms and decision-making basis to operation and maintenance personnel, and edge system-level reliability testing and evaluation frameworks, including multi-dimensional "stress testing" mechanisms such as hardware failures, software crashes, and network jitter.

4.7 Standardization and Interoperability

Currently, most edge computing platforms and algorithms rely on proprietary solutions from manufacturers or research teams, lacking unified standards. In the future, the adoption of open interfaces and protocol standards should be promoted to facilitate seamless integration and ecological sharing among different edge nodes, cloud platforms and algorithm libraries. The reference architectures and evaluation index systems should also be established, and formulate quantifiable comprehensive assessment standards such as real-time performance, accuracy rate, energy consumption and cost, to provide a unified benchmark for both academia and industry.

5 Conclusion

Edge computing, as an emerging distributed computing paradigm, has demonstrated great potential in the field of machine signal processing and fault diagnosis driven by the Internet of Things by moving data processing and decision logic closer to the network "edge" devices that are proximate to data sources. Compared with the traditional model that relies on centralized computing in the cloud, edge computing effectively alleviates the bottleneck of data transmission bandwidth and the shortage of computing resources on the cloud side, and significantly reduces the risk of fault response lag caused by network latency. Specifically, edge nodes can filter, denoise, and pre-extract features from raw sensor data such as vibration, acoustic, and current at the collection end, and then combine lightweight deep learning models (such as the MobileNet series) to complete real-time fault classification and predictive maintenance decisions. This approach not only reduces the decision latency to the millisecond level but also significantly reduces the energy consumption of cloud storage and communication through localized processing, thereby significantly enhancing the agility and overall reliability of industrial Internet of Things systems in responding to sudden faults. At the same time, with the help of dedicated hardware platforms such as STM32 and Edge TPU, as well as soft and hardware collaborative optimization methods, complex neural network models can be efficiently deployed in resource-constrained embedded environments, providing a feasible solution for traditional online, non-intrusive fault diagnosis that is difficult to achieve.

In the future, with the popularization of high-efficiency neural network accelerators (Edge TPU/NPU), the maturity of federated learning and differential privacy technologies, and the introduction of adaptive online learning mechanisms, fault diagnosis systems based on edge computing will evolve towards a "lighter, faster, safer, and smarter" direction, achieving refined, interpretable, and autonomous health management of the entire life cycle of industrial equipment.

References

1. Ganga, D., and Ramachandran, V.: IoT-Based Vibration Analytics of Electrical Machines, *IEEE Internet of Things Journal*, 5(6), 4538-4549 (2018).
2. Liu, P., Zhang, Y., Wu, H., Fu, T.: Optimization of Edge-PLC-Based Fault Diagnosis with Random Forest in Industrial Internet of Things. *IEEE Internet Things J.* 7, 9664–9674 (2020)
3. Sitton-Candanedo, I., Alonso, R. S., Corchado, J. M., Rodriguez-Gonzalez, S.: A review of edge computing reference architectures and a new global edge proposal, *Future Gener. Comput. Syst.*, 99, 278–294(2019).
4. Xiang, S., Qin, Y., Luo, J., Wu, F., Gryllias, K.: A concise self-adapting deep learning network for machine remaining useful life prediction, *Mech. Syst. Signal Process*, 191, 110187 (2023)
5. Shi, W. S., Cao, J., Zhang, Q., Li, Y. H. Z.: Edge computing: Vision and challenges, *IEEE Internet Things J.*, 3(5), 637–646 (2016).

6. STEVAL-STWINKT1B., <https://www.st.com/en/evaluation-tools/steval-stwinkt1b.html>, Jan. 2023.
7. Rubes, O.; Chalupa, J.; Ksica, F.; Hadas, Z.: Development and experimental validation of self-powered wireless vibration sensor node using vibration energy harvester, *Mech. Syst. Signal Process.* 160, 107890 (2021)
8. Wang, X., Lu, S., Huang, W., Wang, Q., Zhang, S.: Efficient data reduction at the edge of Industrial Internet of Things for PMSM bearing fault diagnosis *IEEE Trans. Instrum. Meas.*, 70, Art. no. 3508612 (2021)
9. Tang, H. S., Lu, S. L., Qian, G., Ding, J. M., Liu, Y. B.: IoT-based signal enhancement and compression method for efficient motor bearing fault diagnosis *IEEE Sens. J.*, 21(2), (2021)
10. W. Yu and P. Lv: An End-to-End Intelligent Fault Diagnosis Application for Rolling Bearing Based on MobileNet," in *IEEE Access*, 9, 41925-41933 (2021)
11. Pham, M.T.; Kim, J.-M.; Kim, C.H.: Deep learning-based bearing fault diagnosis method for embedded systems, *Sensors* 2020, 20, 6886
12. Yao, D.; Li, G.; Liu, H.; Yang, J.: An intelligent method of roller bearing fault diagnosis and fault characteristic frequency visualization based on improved MobileNet V3, *Meas. Sci. Technol.* 2021, 32, 124009
13. W. J. Xiao, H. Huang, Y. Sun, Q. Yang: Promise of embedded system with GPU in artificial LEG control: Enabling time–frequency feature extraction from electromyography, *Annu Int Conf IEEE Eng Med Biol Soc.* 2009
14. L. Stankovic and M. Dakovic: On a gradient-based algorithm for sparse signal reconstruction in the signal /measurements domain, *Math. Probl. Eng.*, vol. 2016, Jun. 2016, Art. no. 6212674
15. Huang, Y.; Liang, S.; Cui, T.; Mu, X.; Luo, T.; Wang, S.; Wu, G.: Edge Computing and Fault Diagnosis of Rotating Machinery Based on MobileNet in Wireless Sensor Networks for Mechanical Vibration Sensors 2024, 24, 5156
16. Huang, Q.; Ding, H.; Effatparvar, M.: Breast cancer diagnosis based on hybrid SqueezeNet and improved chef-based optimizer, *Expert Syst. Appl.* 237, 121470(2024)
17. Yang, H.; Liu, J.; Mei, G.; Yang, D.; Deng, X.; Duan, C.: Research on real-time detection method of rail corrugation based on improved ShuffleNet V2, *Eng. Appl. Artif. Intell.* 126, 106825 (2023)
18. Park, D.; Kim, S.; An, Y.; Jung, J.-Y. LiReD: A light-weight real-time fault detection system for edge computing using LSTM recurrent neural networks, *Sensors* 18, 2110 (2018)
19. S. Lu, R. Yan, Y. Liu, and Q. Wang: Tachless speed estimation in order tracking: A review with application to rotating machine fault diagnosis, *IEEE Trans. Instrum. Meas.*, 68(7), 2315–2332, (2019)
20. Crocioni, G.; Pau, D.; Delorme, J.-M.; Grusso, G.: Li-ion batteries parameter estimation with tiny neural networks embedded on intelligent IoT microcontrollers, *IEEE Access* 8, 122135–122146, (2020)
21. T. Verstraeten et al.: Edge computing for advanced vibration signal processing, in *Proc. Surveillance Vishno AVE Conf.*, (2019).
22. S. Lu, J. Lu, K. An, X. Wang: Edge Computing on IoT for Machine Signal Processing and Fault Diagnosis: A Review, in *IEEE Internet of Things Journal*, 10(13), 11093-11116, (2023)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

